Hindawi Publishing Corporation
Journal of Computer Networks and Communications
Volume 2015, Article ID 790375, 24 pages
http://dx.doi.org/10.1155/2015/790375

Research Article

Autonomous QoS Management and Policing in Unmanaged Local Area Networks

Christopher Köhnen,[1] Christian Überall,[1] Muttukrishnan Rajarajan,[1] Rudolf Jäger,[2] and Veselin Rakočević[1]

[1]*City University London, Department of Electrical and Electronic Engineering, London EC1V 0HB, UK*
[2]*Department for Information Technology, Electrical Engineering & Mechatronics, Technische Hochschule Mittelhessen-University of Applied Sciences, Wilhelm-Leuschner-Straße 13, 61169 Friedberg, Germany*

Correspondence should be addressed to Christopher Köhnen; christopher.koehnen.1@city.ac.uk

Received 29 July 2015; Accepted 28 October 2015

Academic Editor: Rui Zhang

The high increase of bandwidth-intensive applications like high definition video streaming in home and small office environments leads to QoS challenges in hybrid wired/wireless local area networks. These networks are often not QoS aware and may contain bottlenecks in their topology. In addition, they often have a hybrid nature due to the used access technology consisting of, for example, Ethernet, wireless, and PowerLAN links. In this paper, we present the research work on a novel autonomous system for hybrid QoS in local area networks, called QoSiLAN, which does not rely on network infrastructure support but on host cooperation and works independently of the access technology. We present a new QoS Signalling Protocol, policing and admission control algorithms, and a new lightweight statistical bandwidth prediction algorithm for autonomous resource management in LANs. This new QoS framework enables link based, access-medium independent bandwidth management without network support. We provide evaluation results for the novel bandwidth prediction algorithm as well as for the QoSiLAN framework and its protocol, which highlight the features, robustness, and the effectiveness of the proposed system.

1. Introduction

Quality of Service (QoS) becomes more and more relevant to consumer networks, due to the increased usage of High Definition (HD) video streaming, IPTV applications, and Voice over Internet Protocol (VoIP) communication. Nowadays, video streaming is already topping the list of consumer traffic [1] with more than 35% of mobile devices using WiFi connections and even Ultra HD Internet video streaming is evolving. Regarding the consumer Internet video traffic, video streaming is expected to grow further massively from 19% as of 2014 to 53% in 2017, as forecasted by Cisco's Visual Network Index [2]. Currently, most consumer routers and switches support QoS management in local area networks (LANs) on the basis of the Differentiated Services (DiffServ). Additionally, there are already consumer systems in the market, which perform self-organised traffic flow identification and classification for prioritised packet scheduling [3]. This means that, at best QoS is only available for traffic passing the gateway and no QoS guarantees are possible as compared to the QoS model the Resource Reservation Protocol (RSVP) is based on [4]. Therefore, LAN internal traffic remains completely unmanaged, which affects mainly intra-LAN communication like video and audio streaming from Media Network Attached Storage (Media-NAS) devices, which is a common operation in networked consumer households. Additionally, IntServ protocols, like the RSVP, usually cannot no be found in consumer network hardware. Even Microsoft has dropped support for RSVP in Windows operating systems with the release of WindowsXP [5] in 2001. State-of-the-art LAN management protocol frameworks like H.622 [6] or UPnP-QoS [7] and QoS NSLP [8] emphasise the need for QoS management in LANs. They propose strategies, which rely on network support for session-based QoS to enable bandwidth reservation.

Since the broadband connection and wireless links in a LAN are still bottlenecks for services with high bandwidth requirements, a QoS scheme is required to protect real-time service flows from congestion. Internet providers solve the problem on the last mile by using separate multicast and QoS enabled Internet links for their IPTV delivery. This provider control ends at the router. Problems in a LAN arise mainly when multiple user traffic flows utilize a bottleneck link in the network and one of the flows has real-time media QoS requirements. This problem in unmanaged network scenarios is still a research challenge in state-of-the-art research and technology. The current trend on application side to use rate-adaptive streaming does not solve the problem either. Rate-adaptive streaming aims at reacting on congestion and bandwidth degradation by stepping down the resolution and bandwidth [9], instead of preventing the congestion from not real-time dependent sources.

To overcome these issues we propose a QoS in local area networks (QoSiLAN) framework, which enables QoS service guarantees in the forms of end-to-end bandwidth reservation between the hosts within the LAN, as well as for traffic flows, passing the Internet gateway. The proposed QoSiLAN framework addresses the problem of QoS management in unmanaged LANs, without the essential dependence on network infrastructure support. Its goal is to manage the real-time traffic autonomously with a host-based, cooperative approach. The QoSiLAN framework is designed to work autonomously, taking into account the assumption not to rely on network cooperation, but to support it. Since the QoSiLAN framework completely relies on host cooperation, multiple key technology modules are needed on the hosts in the LAN to support this architecture as depicted in Figure 1. For the first time, this paper describes all modules and their interaction algorithms in detail.

2. Contribution

In this section the main contributions of our research are presented and discussed. These are the statistical class-based bandwidth prediction (SCBP) algorithm and the QoSiLAN policing and admission modules. These two key solutions provide the core functionality to the QoSiLAN framework, since the bandwidth prediction is essential to support autonomous traffic management, which works independently of support from media applications. Additionally, the policing and admission control module processes all data from the other key technology modules to enable smart policing decisions. A complete overview is shown in Figure 1, where all modules are presented. Especially for self-organisation, the physical topology map of the LAN must be discovered, including a bottleneck bandwidth detection for the path, which is assisted by the iperf tool [10] (topology discovery). Also, the traffic must be monitored, analysed, identified, classified (Flow Identification and Classification), and predicted in an autonomous manner (Flow Bandwidth Prediction). For signalling, a new protocol is needed to enable QoS management communication between the hosts (QoS Signalling Protocol). For this purpose, a signalling solution built on top of the latest IETF recommendation is presented, including a critical examination on the overhead and scalability issues this approach causes. The policing and admission control enables the smart provisioning of resources, which employs cooperative traffic shaping (traffic shaping) on all hosts in the LAN to enforce bandwidth preservation (policing/admission control). In the following, these modules are presented in detail.

2.1. Topology Discovery. First, to manage the traffic on a per link basis, a complete map of the local network must be known. The map reveals the link layer interconnection between hosts, switches, and access points, which is needed for link based resource reservation. The QoSiLAN framework does not rely on network core element support. This means that only end-hosts are involved in the signalling and enforcement processes. Although the network is not involved in the QoS management, the network is actively monitored. To be able to control the traffic in the LAN, its physical topology must be known. This enables the QoSiLAN framework to manage resources link-based and to react on topology changes and congestion directly.

This is accomplished using the Link Layer Topology Discovery (LLTD) protocol. The LLTD protocol was presented firstly by Black et al. [11]. It is a layer 2 topology discovery protocol, which was developed within the Microsoft Rally program [12]. Later the LLTD protocol was introduced and distributed with Microsoft Windows since 2006. The protocol specification was published by the Microsoft Corporation [13]. It works on wired IEEE 802.3 and wireless IEEE 802.11 networks. It is based on probing packets sent by a central entity, called "mapper," to each LLTD supporting host, called "Responder," in the network. All end devices implementing the protocol appear in the network topology map the mapper builds. The protocol aims at investigating the address tables of Ethernet switches. Since one cannot read the Address Information Table (AIT) in most consumer switches, the mapper sends Emit packets to the hosts to make them send Probe packets to the switches using spoofed MAC addresses, according to the connections reasoning technique presented by Sun et al. [14]. Using this algorithm the mapper trains and probes each switch's address tables to find out if the hosts are attached to the same or different switches. According to the IEEE 802.3 specification a switch forwards packets with an unknown MAC address to all ports. Once it has received a packet with the MAC address as source one time it subsequently forwards incoming packets with this MAC as the destination address only to this single port. The other hosts in the network, which have also listened to the training packet also answer to the mapper if they also have received the packet or not. For this purpose, the LLTD protocol uses MAC address spoofing with addresses starting with `00-0D-3A`. This is a dedicated MAC address range, reserved by Microsoft. In order to make use of the LLTD capabilities, we completely reimplemented the protocol and the mapper logic and integrated it into the QoSiLAN framework.

2.2. Flow Identification and Classification. To enable the QoSiLAN to work independently of applications, the traffic flows must be identified and classified in an autonomous

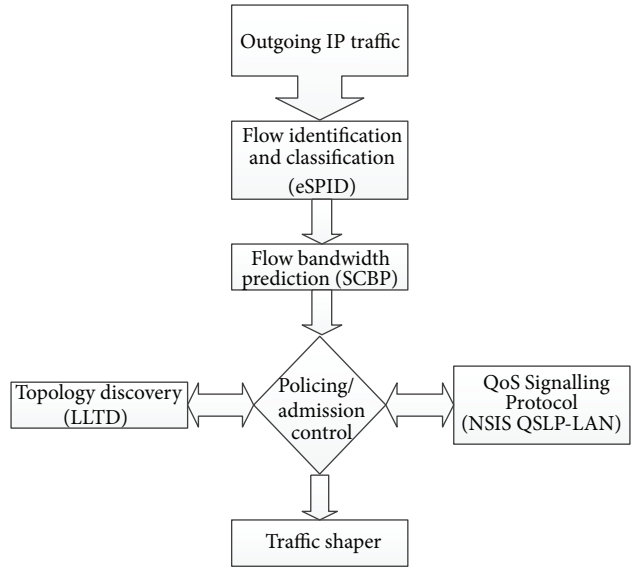


Figure 1: Overview of the QoSiLAN key components.

manner. This is accomplished using our solution, the enhanced Statistical Protocol Identification (eSPID) algorithm [15]. We further developed the Statistical Protocol Identification (SPID) algorithm [16] to identify audio/video streams reliably. The SPID algorithms are based on the Kulback-Leibler divergence (KLD):

$$D(P \parallel Q) = \mathrm{KL}(P, Q) = \sum_{x \in X} P(x) \log_2 \frac{P(x)}{Q(x)}, \quad (1)$$

using twelve different statistical measures of traffic behaviour, like bytes-per-direction, number-of-direction-changes, byte-frequency, and nine more. The eSPID algorithm needs to learn a protocol from 30 preclassified sample flows, as evaluated in our previous work [15]. The twelve statistical measures are applied to the sample flows and stored as probability distribution arrays in a database. This makes it robust against UDP/TCP port changes and up to a certain degree against encryption. The KLD in (1) is a logarithmic measure of the relation between the relative frequency of the observed (P) to the trained flows (Q), summed for each attribute measure. The $D(P \parallel Q)$ is matched to a database of learned $D(P \parallel Q)$ from other protocols. The protocol with the smallest divergence $D(P \parallel Q)$ is then identified. The distance represents the probability of matching. The eSPID algorithm allows for robust identification of flows after 20 packets, as shown in our previous work [15]. Therefore, it is a fast and reliable method for near real-time media identification. The outgoing traffic is identified and classified using the individual protocol name and the media type. These media types are, namely, audio, video, or unknown if no matching was found. New real-time protocols can be learned by the system in a semisupervised way. Needed are at least 30 sample flows, which are identified by the user. For these the KLD is calculated and the result is stored in the database. The identification probability of encrypted protocols is not substantially less. As the content based measures have no statistical information due to the entropy of the encrypted content, only the behaviour focused statistical measures apply. This does not need to essentially lower the recognition probability, especially not for different protocols, but for the content differentiation within a protocols.

2.3. Statistical Class-Based Bandwidth Prediction. Once a flow is identified and classified, the needed resources must be estimated in order to reserve them. This is enabled by a novel statistical class-based bandwidth prediction (SCBP) algorithm, which we developed for the special purposes of the QoSiLAN framework. As shown in Figure 1, it is applied after the eSPID process and the results are fed directly into the policing and admission control module.

The statistical class-based bandwidth prediction (SCBP) algorithm is our novel approach for Internet video traffic bandwidth prediction for single streams. It addresses the problem of different transmission characteristics of state-of-the-art streaming technology and the resulting low bandwidth forecasting accuracy. Nowadays, the traditional continuous streaming characteristic, for example, known from the Real-Time Transport Protocol (RTP), is highly underrepresented. Instead, HTTP based streaming technology like HTTP Dynamic Streaming (HDS) [17], HTTP Live Streaming (HLS), and the Real-Time Media Protocol (RTMP) is very common. Those streaming protocols support the feature of chunked data transfer. This allows the CDNs to significantly save their traffic costs, since only those parts of a video are transferred, which are actually consumed by the user. If a user stops a video the transfer also stops. If a user moves to a position at the end of a video, there is no need to transfer the parts before. So, the data is not transferred continuously according to the encoded bit rate, but in chunks, with a burst characteristic, often with several seconds of traffic silence in between the bursts. The goal for our SCBP algorithm is to predict the average bandwidth requirement of a flow, for the

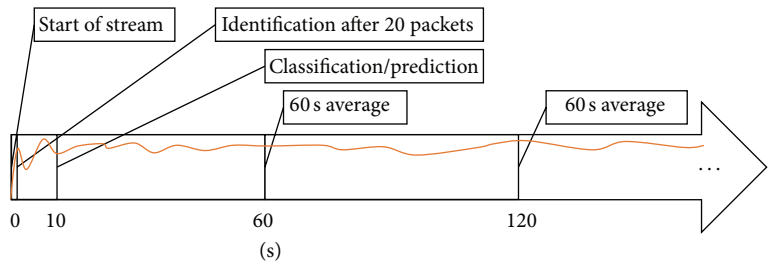


Figure 2: Bandwidth prediction procedure timeline.

first minute of transfer, ten seconds after start. As shown in Figure 2, after 60 s the average of the last 60 s is chosen to update the reservation state. These values were chosen to fulfil system requirements for the QoSiLAN framework. So, after 60 s the reservation update is not using the SCBP values, but the measured average throughput of the flow from the last 60 s.

Since the QoSiLAN framework needs to communicate with all hosts in the network to coordinate the resource reservations, it causes high signalling effort. Therefore, the reservation state live time was evaluated to be 60 s, to avoid too frequent state updates and therefore network congestion from signalling. That is why the SCBP algorithm is designed to predict the average bandwidth consumption for single multimedia streams for the first 60 s after 10 s recording time. According to our evaluations a minimum of 10 s recording time from the start of the flow was evaluated to be required to catch the flows characteristic and to enable proper predictions. As depicted in Figure 2, after 60 s of measurement, the reservation state is updated to the measured average bandwidth consumption ($\overline{B_{60}}$) of the last 60 s. Within this paper we present our simple but highly precise prediction algorithm and validate its applicability using evaluations, as presented in Section 5.1.

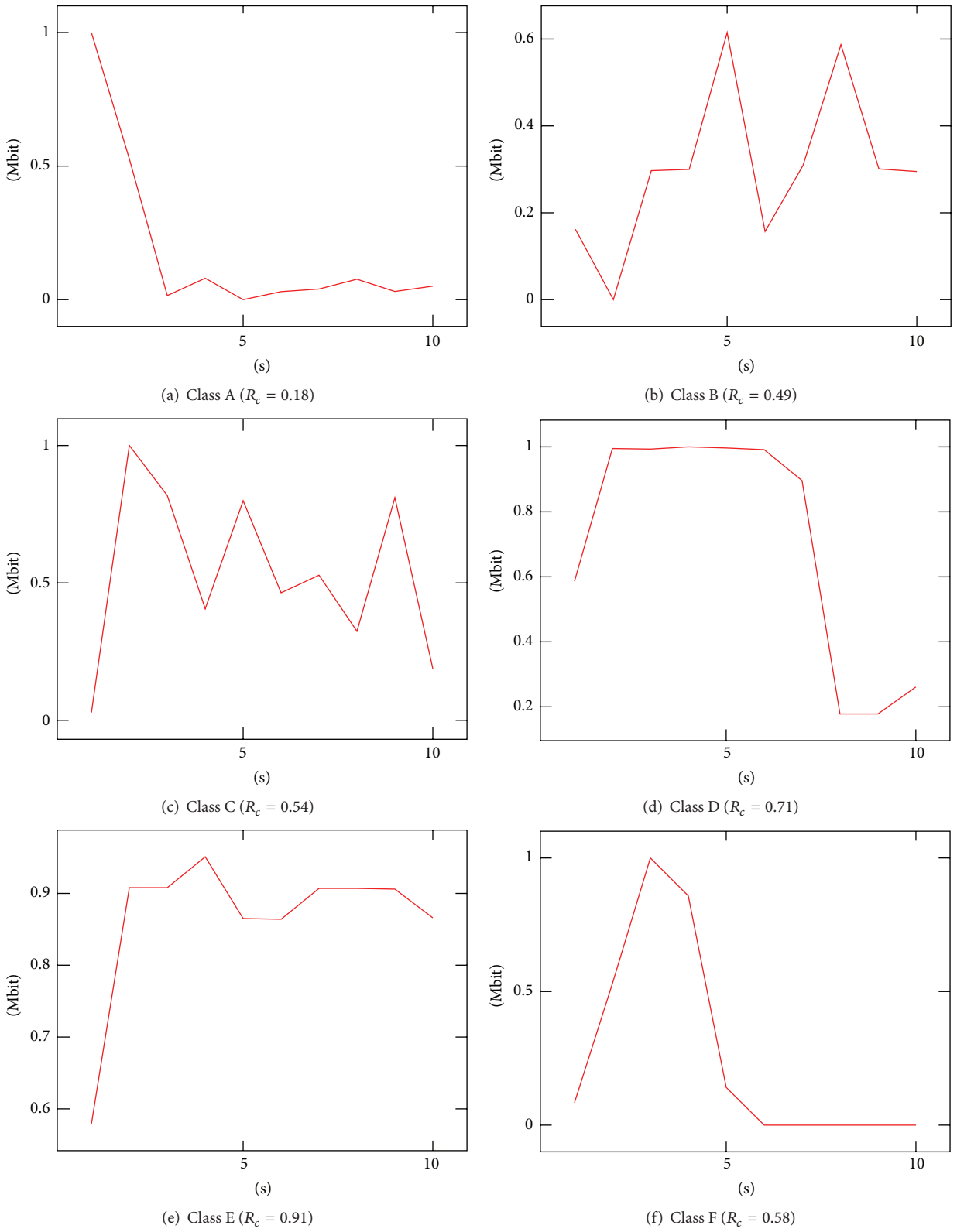
The SCBP algorithm classifies the traffic into six different traffic classes (A–F), according to their behaviour within the first ten seconds of transmission, as shown in Figure 3. To achieve the highest accuracy, the measurement must start at the very beginning of a connection. During the observation time I m throughput measurement samples χ_i are collected, where m is an even number. From this we get the set B, containing m bandwidth measurement samples. For better flow characterisation we divide the set of B into two subsets: $B_a \subset B := \{B_0, \ldots, B_{m/2}\}$ and $B_b \subset B := \{B_{m/2+1}, \ldots, B_m\}$, where we calculate the maximum ($B_{a,\max}$ and $B_{b,\max}$) as well as the average ($\overline{B_a}$ and $\overline{B_b}$) values. In addition also $B_{\max}$ is determined as the maximum value in the set of B. The prediction is calculated at time I and forecasts the average bandwidth consumption for the whole first 60 s of transmission. The sampling interval is defined as $s = I/m$. The evaluations are based on $s = 10\,\text{s}/10\,\text{samples} = 1\,\text{s}$. The ratio R_c, defined in

$$R_c := \frac{\overline{B}}{B_{\max}}, \tag{2}$$

Table 1: Traffic classes by characteristics.

Class name	Characteristic (R_c)
A	$0 \le \frac{\overline{B}}{B_{\max}} \le 0.25$
B	$0.25 \le \frac{\overline{B}}{B_{\max}} \le 0.5$
C	$0.5 \le \frac{\overline{B}}{B_{\max}} \le 0.6$
D	$0.6 \le \frac{\overline{B}}{B_{\max}} \le 0.8$
E	$0.8 \le \frac{\overline{B}}{B_{\max}} \le 1$
F	$\frac{\overline{B}}{B_{\max}} < 1 \wedge \overline{B_b} = 0$

classifies the flows into the categories A–F, as shown in Table 1. There, especially the ratio of bandwidth consumption within the first five and the next five seconds is taken into account. The classes represent dedicated constant values, which are applied to k_c in (3), the prediction algorithm. The SCBP algorithm predicts the bandwidth requirements for identified streams for the period of 60 seconds after 10 seconds of observation. These values come from the QoSiLAN framework requirements, which were defined to require a detection and prediction of QoS relevant flows within 10 seconds after they start. Additionally, to reduce the amount of signalling messages, QoS state updates should not be sent more frequently than each 60 seconds which requires for a prediction interval of 60 seconds. Table 1 shows samples for the characteristics in the first ten seconds for the different traffic classes. One can see that the criteria basically reflects the first burst length, which is typical for nowadays Internet streaming. With the first burst the first chunks of video frames are pushed to the client to fill the client video buffer for enabling a fast start of video playback. For that reason the first burst provides a first estimate on the expected average throughput rate. To enhance the estimate, the next bursts are analysed with the set of B_b. The class F addresses the special characteristic, when there are no samples in the set of B_b, but the stream has not finished yet. Figures 3(a)–3(e) show samples for the first 10 s characteristic of the traffic classes. There, Figure 3(e) represents the classical continuous streaming case, whereas Figure 3(f) represents the case, with



(a) Class A ($R_c = 0.18$)

(b) Class B ($R_c = 0.49$)

(c) Class C ($R_c = 0.54$)

(d) Class D ($R_c = 0.71$)

(e) Class E ($R_c = 0.91$)

(f) Class F ($R_c = 0.58$)

Figure 3: Examples for the SCBP classes.

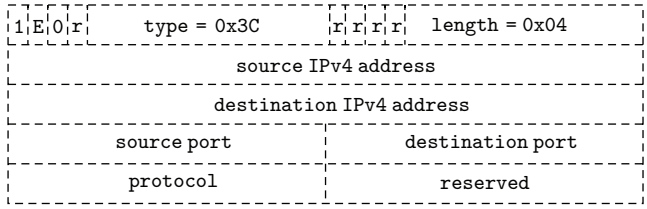


Figure 4: Reservation path parameter for IPv4.

a burst at the beginning and no data within the last half of the observation interval of 10 s.

Through evaluations, we validated that the k-values assigned to the classes A–F were applicable to the ranges as best defined in Table 1. Hence, the classes reflect the bandwidth characteristic, including the start-up behaviour of the flow in a simple way. For the class-range assignment evaluations, the measured error values were put into a density histogram to identify the ranges of similar deviations.

We designed the prediction formula, shown in

$$\overline{P_{10}} = \frac{\left((1/N_{10})\sum_{i=0}^{N_{10}}\chi_i\right)^2}{(1/N_5)\sum_{i=0}^{N_5}\chi_i} * k_c = \frac{\left(\overline{B}\right)^2}{\overline{B_a}} * k_c, \quad (3)$$

for the balance of simplicity and accuracy. It takes into account the application in embedded systems with less computation power, which also saves battery life on mobile systems. The prediction formula in (3) implements the correction factor k_c, which reflects the class characteristics. The value for k_c was found using evaluations, as carried out in Section 5.3.

2.4. QoS Signalling Protocol. The QoSiLAN Signalling Layer Protocol (QSLP-LAN) provides the mechanism to exchange QoS information and signalling commands within the QoSiLAN hosts (QH) on top of the NSIS NSLP framework, adapted for the QoSiLAN QoS model. As as shown in Figure 1, the protocol is employed by the policing and admission control module to coordinate the QoS policies within the hosts in the network. The QSLP-LAN is designed according to the latest IETF Next-Steps-in-Signalling (NSIS) recommendations [18, 19], and named QoS Signalling Layer Protocol for QoSiLAN (QSLP-LAN). The core piece of the QoSiLAN approach is the policing and signalling procedures. Whereas present QoS protocols communicate end-to-end and QoS aware network elements where involved, in QoSiLAN it is different. One host, which preferred the originator of the traffic, informs a QoSiLAN Manager (QM) host, about the needed resources. A QM could be any host in the network. Using the network topology map, the QM generates sophisticated policy resource requests and sends them to all other hosts in the network to indirectly achieve resource reservation for the physical links along the data path. In the following, we present the QSLP-LAN protocol messages and the signalling procedures. The protocol behaviour within the QoSiLAN framework is presented as examples within the evaluations in Section 5.2.

2.4.1. QSLP-LAN Message Format. The QoS Signalling Layer Protocol for QoSiLAN is designed based on the NSIS message format, similar to the QoS NSLP header format described in [19], Section 5.1.

The QoSiLAN protocol uses, like other common QoS protocols, a soft state mechanism. This means, the hosts keep a reservation state as long as a time-out has not been reached. To keep states alive, a reservation refresh message (QoSiLAN_RESERVE), with the same parameters as the first one, must be sent before the time-out is reached.

QoSiLAN hosts (QH) inform the QM about their traffic situation and request for resources. The QM asks all QH for cooperation for currently active QoS states to establish the bandwidth reservations commonly.

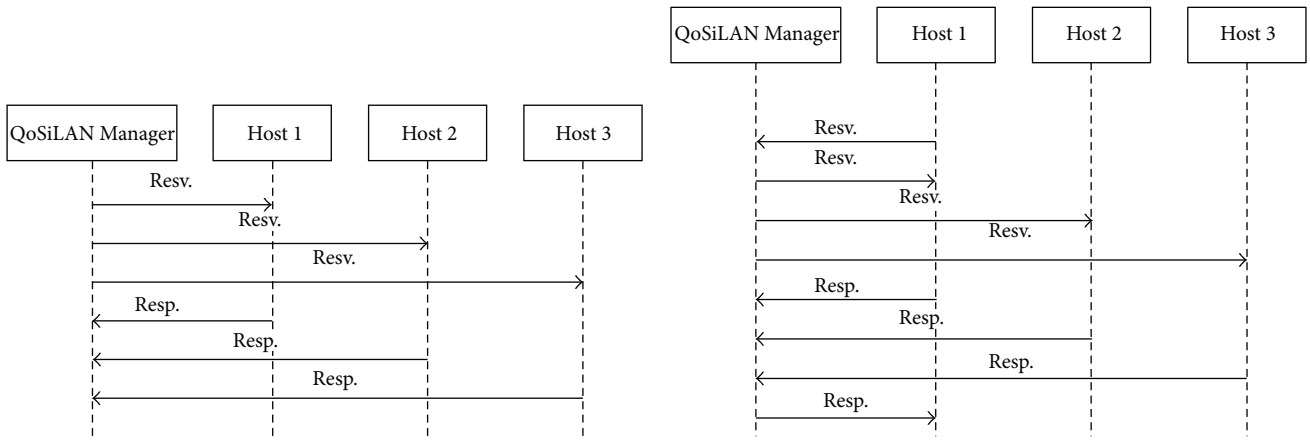
The NSIS General Internet Signalling Transport (GIST) protocol [20] also brings along a host discovery feature, which is used to detect and identify the QoSiLAN enabled hosts in the network. The QoSiLAN Manager (QM) selection is performed by the smallest switch/hub-hop distance to the router, evaluated using the LLTD protocol. If the router itself is QoSiLAN enabled, it announces a distance of zero. If the router is not QoSiLAN enabled and two QH have the same distance, the QH with the longest uptime period is selected.

(i) Reservation (QoSiLAN_RESERVE). The reservation messages use the NSLP common header, as all GIST NSLP objects [20] do. The message contains a sequence number (RSN), a REFRESH_PERIOD, and a BOUND_SESSION_ID. The QSPEC object defines the QoS information parameters. It lists the requested resources using the `TMOD-1` parameter, which contains the peak data rate for traffic shaping, and the `Excess Treatment` parameter, which defines the shaping policy, as defined in [21].

In addition, we introduced the new `Reservation Path Parameter for IPv4`, to communicate the five-tuple, which defines what flow the reservation is for, as shown in Figure 4. The first 32 bits are defined according to the QSPEC parameter header [21]. The source and destination IPv4 addresses as well as the source and destination ports and the transport protocol ID are included. For IPv6 a similar parameter header format with 64-bit IP address fields needs to be defined accordingly.

Reservations are deleted either on soft state time-out or by sending a QoSiLAN_REQUEST message using the BOUND_SESSION_ID without a QSPEC object defined.

(ii) Response (QoSiLAN_RESPONSE). The response message is intended to report success or error codes to the requesting



(a) QM initiated message flow

(b) Receiver initiated message flow

Figure 5: QM and receiver initiated message flow.

node. The success case is indicated using the `error=0x00` value. If a reservation state is rejected, a negative acknowledgement is signalled through a `error=0x47` value set in the INFO_SPEC header. The RSN object is taken from the corresponding QoSiLAN_REQUEST message to match the response to a request.

(iii) Notify (QoSiLAN_NOTIFY). The notify message is intended to report significant best effort traffic flow statistics to the QoSiLAN Manager (QM). This is needed to keep the QM informed about the current state of the network, especially when there is a lot of traffic occupying which is not classified for QoS. The message structure is similar to the corresponding QoSiLAN_REQUEST and QoSiLAN_RESPONSE messages as described before. The QoSiLAN_NOTIFY message also uses the Reservation Path Parameter for IPv4 within the QSPEV header to announce detected flows.

2.4.2. QSLP-LAN Signalling Procedure. In Figure 5(b) the signalling is shown to be exemplary and used to establish bandwidth reservation in a simple LAN scenario. In this LAN one QoSiLAN Manager (QM) host and three client QHs are connected using two different switches. All of them, except the switches, are QoSiLAN aware. The initiator (Host 1) sends a reservation request message (QoSiLAN_RESERVE) to the QM. The reservation request shall contain at least the physical addresses and the IP addresses of the two communicating parties and the requested resource, the predicted bandwidth to reserve. The QM analyses the location of the QHs and sends sophisticated reservation requests (QoSiLAN_RESERVE) for all other nodes (Host 1, Host 2, and Host 3) in the network, based upon the LAN topology and the LAN's traffic status knowledge, to encourage all QHs to obey the limitations for the affected physical links. In return, the nodes acknowledge the request to the QM (QoSiLAN_RESPONSE), and the QM reports (QoSiLAN_RESPONSE) the result/success to the initiator (Host 1) at the end. To cover the case where both the sending and receiving host do not support QoSiLAN, but the QM is a gateway and detects the flow, another example is given. As shown in Figure 5(a), on resource reservation initiation by a gateway, enabled as QM, no request/response communication to another initiation node is required in contrast to the host initiation case, depicted in Figure 5(b). In both cases the reservation requests/responses are sent to/from each QH in the network, initiated from the QM. To avoid unnecessary signalling effort, the hosts do not report their traffic statistics regularly, but only triggered by events. In case of significant best-effort traffic detection, a QH reports the monitored bandwidth of the stream to the QM using a QoSiLAN_RESPONSE message. Details on the policing and admission control algorithm are described in Section 2.5.3.

2.5. Policing/Admission Control. As shown in Figure 1, the policing and admission control algorithms are the key components to enable autonomous QoS management. Within this paper we propose appropriate policing and admission control schemes to enable the QoSiLAN framework. The resource management within a LAN requires detailed knowledge about the available resources on all links between the LAN entities. Only if the link layer topology and the capacities of its links are known, the QoSiLAN framework can autonomously manage the resources. In the following, we describe the management procedures to acquire the required information.

2.5.1. Resource Discovery. One host in the network, which preferred the gateway, fulfils the role of the QM. As such it maps the network topology and acts as resource coordinator. The mapping process is executed each time a new host is discovered in the network. The host discovery is based on broadcast and ARP packet monitoring. The map, generated by the LLTD Mapper module, described in Section 2.1, reflects the link layer topology between the hosts, switches, hubs, and access points in the LAN. Additionally to the topology itself, the bottleneck capacity of each link needs to be evaluated. This is done during network or link idle times.

The mapper advises the hosts to measure the bottleneck bandwidth of their links by active probing. The QM gathers the results from the QHs and adds this information to the connection information table within the topology map.

2.5.2. Policing Procedure. Each QH in the network continuously monitors its outgoing traffic. To make autonomous QoS policing possible, the QM needs to gather and maintain all information about the LAN and its links, by using the LLTD topology mapper module. This includes the physical paths and measures their bottleneck bandwidth μ_l. As depicted in Figure 6, the eSPID module analyses on each QH the outgoing flows to identify data with QoS requirements, like video and audio transport. Once a host discovers, for example, a VoIP communication, it estimates the bandwidth requirements P_B for this particular flow using the SCBP algorithm. The QoSiLAN MBAC algorithm decides upon the admission of the reservation, taking into account the different measures. If the admission was granted, a *QoSiLAN_REQUEST* message is sent to the QM to request the bandwidth reservation for the detected resources. A *QoSiLAN_REQUEST* message contains a five-tuple: the sender and receiver IP and port addresses as well as the estimated bandwidth.

The QM receives the *QoSiLAN_REQUEST* message and checks the map and the available resources for the flow's route within the LAN. If the requested resources are available, the QM sends individual *QoSiLAN_REQUEST* messages to each QH in the network. These messages contain a list of QSPEC parameters containing the flow's five-tuples and their bandwidth limits to be obeyed by the receiver. These messages are generated individually for each QH, since the affected links on a route to another QH in the LAN differs, depending on the location in the LAN's topology. Each QH checks the request for validity and if the requested resources are available locally. In any instance of an error or conflict, the QH sends a negative acknowledgement *QoSiLAN_RESPONSE* message back. If the request is accepted, the QH sends a positive acknowledgement *QoSiLAN_RESPONSE* message back to the QM.

Once all QHs in the network responded with a positive acknowledgement *QoSiLAN_RESPONSE* message, the QM confirms the resource request with a positive acknowledgement *QoSiLAN_RESPONSE* message to the requesting QH. From this moment, the flow is protected by traffic shaping rules on all QHs in the network, which apply to all flows, except the protected one. The reservation state is deleted either on soft state time-out or on end-of-stream discovery, like TCP FIN flag detection.

2.5.3. Admission Control. The proposed admission control algorithm works according to the principles of coordinated resource admission control. This shall prevent overbooking of resources. The QM provides the function of a final decision point. It carries out priority considerations in terms of network resource availability, based on client requests. It also takes care that resource reservation request does not block best effort signalling traffic and that individual links are not overbooked. According to comparative simulation results from Jamin et al. [22] a network utilisation rate of 77% is achievable in a multihop scenario, with a utilisation target of $80\% = 1 - \mu_r$ and no packet loss due to congestion. For this, we defined a threshold of $\mu_r = 20\%$ residual capacity, which shall not be blocked by reservations and is instead reserved for best-effort traffic. The following algorithm defines that a reservation will be denied by the QM:

$$\mu_l \geq \mu_c = p_b^{\alpha} + \mu_r + \sum_{i=1}^{N} \widehat{\nu}_i, \tag{4}$$

if the sum of current reservation states $\widehat{\nu}$ for one of the affected connections/links including the predicted bandwidth p_b and residual capacity μ_r for a flow α exceeds the link capacity μ_l.

This algorithm enforces bandwidth allocation by collaborative shaping. By default, the QHs in the network shape their traffic to the residual bandwidth, to isolate the reservation's traffic from congestion. This works fine for a two-host network. The probability of multiple hosts using the full capacity of residual bandwidth and therefore exceeding it in sum grows with every host joining the network. Therefore, an additional control function is needed to manage the residual bandwidth and to share this resource for the best effort traffic among the hosts. A reactive approach, as proposed by Hock et al. [23] cannot be applied to this scheme, since the traffic characteristic is not predictable enough. As carried out in Section 2.3 the traffic characteristic is not continuous, but bursty with pause periods. Therefore, a QoS degradation by congestion is hard to detect by the receiver. Hence, our approach works as follows.

Since every QH in the network is monitoring its outgoing traffic continuously, it detects streams with significant best effort traffic rates. The rate is regarded as significant, if the output rate r_o exceeds the residual capacity shared by the number of QHs $r_o \geq (1/n)\mu_r$. If this is the case, the QH informs the QM about its current average best-effort traffic output rate and the destination of the stream. The QM collects this information from all QHs and maintains it per link in the network topology map. The QM takes care, that the sum of best-effort traffic from all QHs will not exceed μ_r for each single link in the LAN. If a reservation violation is detected by the QM, the affected QHs are advised to shape their traffic accordingly for traffic crossing the affected links.

3. Related Work

In this section, related work for the new technology proposed in this paper is discussed. These are the QoS management in LANs, the statistical class-based bandwidth prediction for flows, and the policing and admission control in LANs.

3.1. Approaches for QoS in Local Area Networks. Beside the QoS approach presented in this work, other solutions for enabling QoS in LANs or home networks are described in the literature. The approaches vary from the access layer to the application layer. Since the QoSiLAN approach is an approach targeting cross access technology scenarios, other approaches working on layer 1 and layer 2 are not discussed.

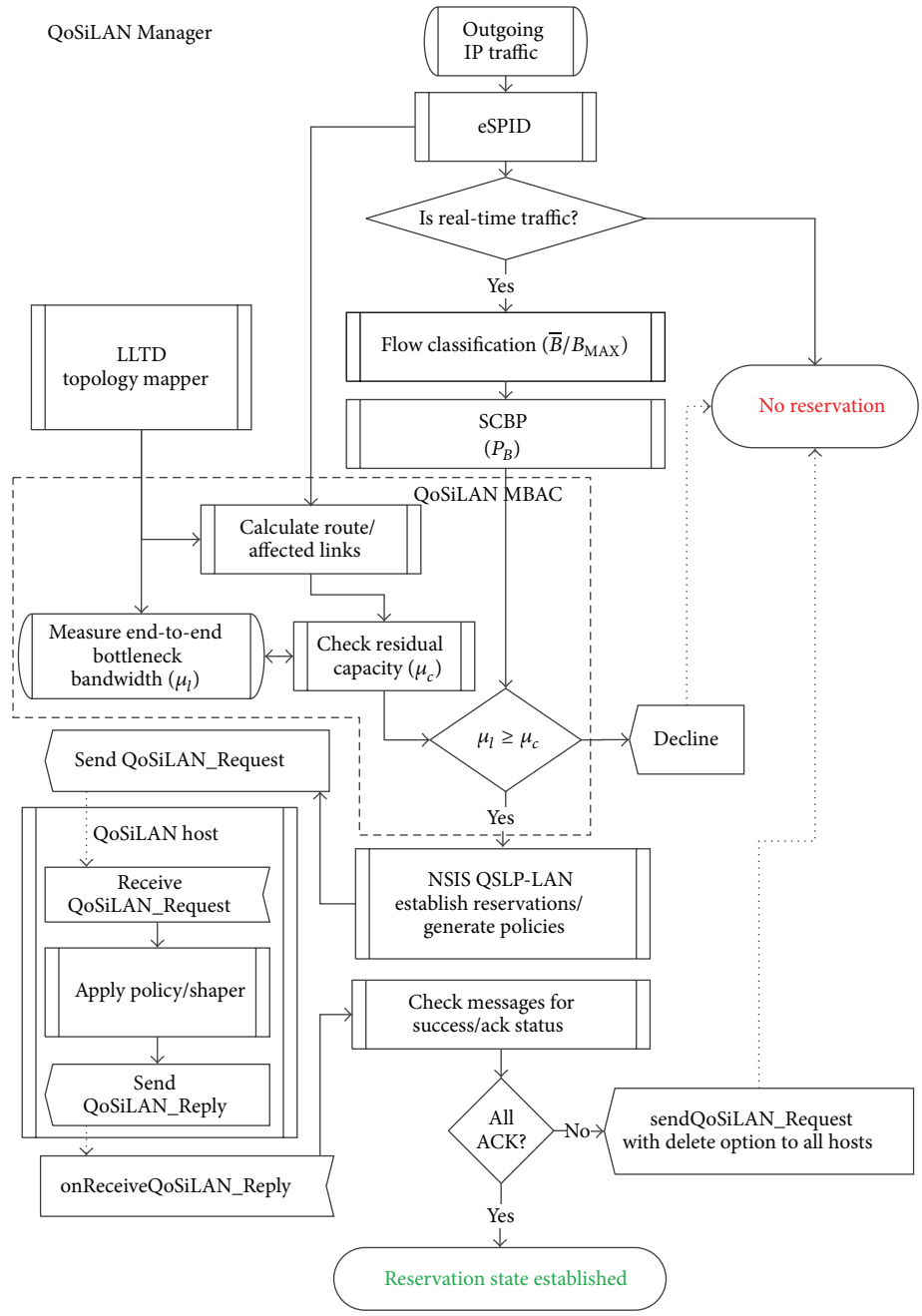


Figure 6: QoSiLAN policing information flow diagram.

An approach, most similar to the QoSiLAN framework, has been proposed by Louvel et al. [24–26], who propose a network resource management framework for multimedia applications distributed in heterogeneous home networks. In this solution to QoS for multimedia applications in LANs, also a central management entity, called the Global Resource Manager (GRM), is used as resource coordinator. On the local devices, the needed components are bundled in a Local Resource Manager (LRM). The LRM provides a resource estimation method, implemented using the Bienaymé-Chebyshev inequality algorithm [27] and a scheduling tool for traffic prioritisation, the Linux iproute2 tool's `tc` command [28]. The GRM measures the available bandwidth on the links using the iperf tool [10] and coordinates the resources. As main differentiation criteria, Louvel's proposal does not take into account the network topology and limits the approach to a one-hop star topology with heterogeneous network interfaces and different devices attached. Since the GRM as the central entity is able to manage all resources, a dedicated QoS protocol is not needed and obviously out of scope of the approach. This limits its practical applicability in home networks dramatically. In the defined topology

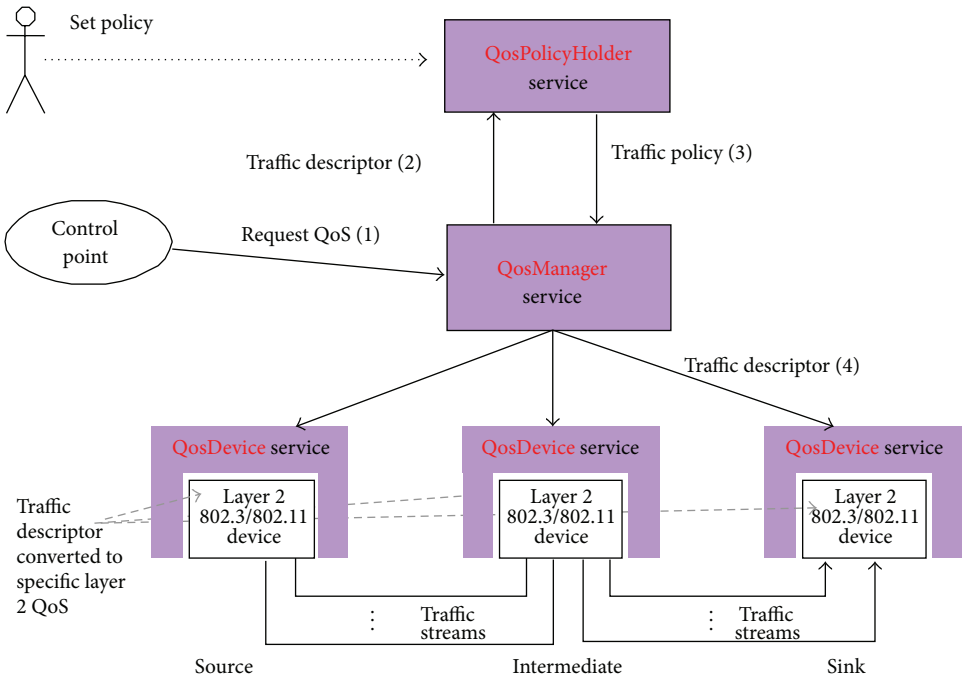


Figure 7: UPnP-QoS architecture averview [7, Figure 1].

it benefits from the nonintrusive and adaptable resource management approach, since the end-devices do not need to be modified essentially to achieve the desired QoS level.

A state-of-the-art approach for a common home network protocol comes from the Universal Plug and Play (UPnP) forum in forms of the UPnP-QoS Architecture [7] to enable QoS services in LANs, consisting of a single IP subnetwork. This architecture defines policing and admission control for prioritised, parametrised, and hybrid QoS control for individual links, as well as path property discovery for them. Three services are required to implement this functionality, as presented in Figure 7. The QosPolicyHolder Service gathers path information and provides appropriate policies for the traffic, described by a TrafficDescriptor structure. The QosManager Service, invoked by an application and implemented within a UPnP Control Point, requests the required resources from the QoSPolicyHolder Service. The QosDevice Service is responsible for establishing the QoS for a new traffic stream. To support end-to-end prioritised QoS, the QosDevice Service needs to be implemented on all network devices along the data path. For links not supporting the parametrised QoS on the path, prioritised QoS is selected hoping for DiffServ support by the network, resulting in an hybrid QoS operation. Network segments not supporting the UPnP-QoS architecture result in a QoS establishment failure for the whole end-to-end path. This is an aspect the proposed QoSiLAN framework overcomes, since it does not rely on network support and it can operate end-to-end, even if not all devices support the system. In addition, the UPnP-QoS framework requires not only implementation of its services on all devices, but also support by the applications causing the traffic. This problem was addressed by Laulajainen and Hirvonen [29], who propose a background service running on traffic causing devices, which performs fast application detection on basis of statistical analysis of the first four packet sizes of a stream, which provides a basic identification functionality. This measure is also a small part of the eSPID algorithm, described for the QoSiLAN framework in Section 2.2. Suraci et al. state that the UPnP-QoS architecture works well in the case of moderate traffic loads but may fail whenever the network becomes overloaded [30]. They demonstrate their admission control and drop solutions using test-bed evaluations. For the admission control algorithm they rely on the bandwidth information provided by the UPnP framework. If a network segment on the data path has no sufficient residual bandwidth, the admission is rejected; otherwise it is admitted. The drop strategy decision algorithm is realized using a binary tree to estimate lowest cost for packet dropping. They determine the cost by the importance/priority of the flow. The QoSiLAN framework also defines an admission control algorithm, which also supports not QoS aware network segments, in contrast to the presented solution. A dedicated drop strategy is implicitly given by the traffic shaping appliance the QoSiLAN requests from the operating system. Castrucci et al. go one step deeper in the scheduling of packets with their proposal for an application QoS management and session control in a heterogeneous home network using inter-MAC layer support [31]. They propose an architectural and procedural definition of the home context using the UPnP-QoS and SIP frameworks. Within this OMEGA called architecture they introduce a convergence layer between IP and MAC layer to manage all traffic using the information provided by the UPnP-QoS framework. This architecture requires implementation on all network devices and also uses a centralised coordinator within

the network's gateway to manage the resources. Chen et al. [32] propose a DiffServ focused scheme for QoS management in heterogeneous home networks. it also adopts to the QoS-UPnP specification by adding monitoring and resource management functionality to the framework. They monitor real-time network traffic to adaptively control the bandwidth and manage to reduce jitter latency and packet loss significantly. The focus of this particular work is set on the last mile from the service provider to the home network, which is out of scope of the QoSiLAN framework. Since the QoSiLAN framework also may control the Internet gateway, it also manages these resources and controls the Internet line. Furthermore, Westberg et al. use the OSGi's resource management in DiffServ (RMD) [33] architecture to interface to the Per Hop Reservation (PHR) and Per Domain Reservation (PDR) protocols to manage the network traffic not on per flow basis, but on link basis. In this way they also cover wired and wireless QoS concerns into admission control using the proposed adaptive QoS mechanism, proved by evaluations in an heterogeneous test-bed. Lee et al. [34] propose an enhanced UPnP QoS architecture to support network-adaptive media streaming in home networks. They state that the initial UPnP QoS architecture does not provide methods for dynamic network monitoring. Thus, they propose to enhance it by adding a dynamic network monitoring and adaptation scheme. Although the UPnP QoS 2.0 specification introduced the GetRotameterInformation method to retrieve network status information of other important features, it still lacks the QoS-based adaptation method and the capability of guaranteeing streaming quality over time-varying networks. They propose to enhance the UPnP QoS Device with an dedicated Status Monitor component and the UPnP QoS Manager with an QoS Adapter. Their purpose is to acquire continuous network status information for a dynamic QoS management. They use this enhanced functionality to adapt the video streaming quality dynamically. This approach is different from the QoSiLAN framework, since it does not aim to prevent congestion, but only to react on network performance degradation, which leads to lower video quality and therefore probably lower QoE. Brewka et al. [35] propose an enhancement to UPnP QoS for automatic QoS provisioning, which is a missing feature to have a better comparability to the QoSiLAN framework. They describe the problem of autoclassification of the traffic from non-UPnP-QoS devices present in UPnPQoS enabled networks. A limitation of their proposal is the assumption that the home gateway and all network devices are UPnP QoS aware and support their enhancements. Only end-devices are allowed to be not UPnP QoS compliant. An advantage is the integration of the provider network using GMPLS transmission, which provides better QoS enforcement possibilities. Their simulation shows similar results as ours with the same setup time issues, which are inherited from the reactive traffic identification and classification approach.

Also the ITU-T proposes an architectural framework of a home network that supports multimedia services within the recommendation H.622 [6]. The ITU-T identifies two different roles that home networks fulfil and name them as primary and secondary domain. For the primary domain the home network is considered as an extension of the access network from the provider point of view. For the secondary domain, they consider the home network as an intra-LAN transmission medium for data distribution among home devices from the user point of view. As an extension of the access network, they state that providers expect it to behave similar to their access network with the same functional QoS services with security and management entities that can be found typically in provider networks. In the role of interconnecting home devices these features may not be needed. For QoS they also define two different QoS models: class-based QoS and session-based QoS. The session-based QoS is recommended to be realized as UPnP [36] and class-based QoS using [37, 38]. They emphasise the features of these models, like for class-based QoS the less complexity, scalability, and priority-based mechanism. For session-based mechanisms they criticise that some network devices may be unaware of signalling protocol, because network devices need a complicated mechanism and that additional session setup time is introduced by the resource reservation process. Interestingly, they also consider NSIS QoS NSLP [19] and UPnP QoS [7] as emerging new QoS technology which need further consideration. This is exactly what also the QoSiLAN framework does by further developing the ideas from NSIS and UPnP to enable autonomous session-based QoS for unmanaged networks. In that way the QoSiLAN framework complies with the H.622 recommendation for the primary as well as the secondary domain and fills the gaps of the identified drawbacks of existing and referenced solutions.

There are also QoS approaches for higher layer application, for example, the routing layer. Haikal et al. propose a distributed QoS adaptive routing engine architecture based on OSPFxQoS [39, 40]. This is an Open Shortest Path First (OSPF) link-state routing protocol extension, which works independently of the used QoS architecture. This kind of routing-level QoS architecture works well for large scale hierarchical, routed networks but does not provide a solution to unmanaged local networks using a single subnet, which is the target environment for to QoSiLAN framework.

The Data Distribution Service (DDS) for real-time systems [41] is a middleware architecture for device, service, and QoS management for data centric communication in highly dynamic distributed networks. It follows the publish-subscribe communication model and is able to provide QoS in any environment, where users, devices, and services are potentially mobile. Al-Roubaiey and Alkhiaty provide an architecture for a QoS aware DDS middleware in an ubiquitous environment [42]. Their proposed solution as well as the DDS specification does not provide technical solutions, but only high level descriptions of solution principles and are therefore not directly comparable to to QoSiLAN framework.

3.2. Bandwidth Prediction for Flows. In the past, the scientific community addressed the problem of traffic prediction mainly to continuous traffic flows, Internet backbone traffic or even more specific on video codec level. The algorithms, designed for encoding bandwidth prediction, use

algorithms to exploit the nature of MPEG video to allocate the bandwidth on a scene basis [43]. The algorithms used for Internet backbone prediction address scenarios, where multiple streams run through one link and predictions aim on providing forecasts for the multiplex, the sum of streams [44]. In contrast, our solution addresses only single stream predictions, in a local network scenario. In publications addressing streaming media, mainly traditional streaming protocols were investigated. To the best of our knowledge there is no comparable scientific publication looking at the nature of real Internet traffic from modern cloud services and Content Delivery Networks (CDNs) on a per flow basis. Even in the area of traffic modelling no comparable work could be found. An often addressed problem is the traffic prediction on a large scale basis for Internet backbones [45], which aims on statistical predictions like number of streams, amount of bandwidth, and occurrence probability. Some general applicable algorithms, like the Recursive Last Square (RLS) in an application of traffic prediction [46] and machine learning algorithms like SVM [47], were investigated. It was found that those algorithms are on one hand designed to predict the traffic on a short term but do not perform very well in an inert framework like the QoSiLAN addresses, with forecasting intervals of 60 s. In addition, those algorithms cause high computation complexity, but the QoSiLAN framework's design requires for a lightweight approach with a minimum of computation costs, since it is designed to run on thin and also mobile systems with limited CPU and power sources. That is why we aim on a simple approach to predict the needed bandwidth as accurate as possible. These requirements are mostly hit by linear prediction algorithms. He et al. [48] distinguish between Formula Based (FB) and History Based (HB) algorithms. For the FB algorithms they propose linear prediction algorithms. Among others a Moving Average (MA) predictor was presented. We also employed and configured it for our application and referenced it as Mean Estimation (ME), for our evaluation of results in Section 5.3. The HB algorithms do not fit for our application, since they require a large set of throughput measurements from previous transfers in the same path, which behaved similar. In particular, the similar behaviour of protocols is an assumption we could not reproduce in our evaluations, where similar video streaming flows from different large CDNs behave very individually.

3.3. Policing and Admission Control in LANs. A framework for providing end-to-end QoS for individual flows was proposed by Yang et al. [49] with the goal of keeping the scalability of the DiffServ model. They propose the On-Demand QoS Path (ODP) framework, which supports per flow admission control and end-to-end bandwidth reservation. In contrast to our QoSiLAN framework, the ODP framework targets interdomain/Internet QoS by involving edge- and core-router. The ODP framework enables scalability by using class-based service differentiation in the network core. The ODP framework reduces the signalling effort by using a hierarchical bandwidth management scheme. From evaluations they conclude that the ODP Central Control and Router-Aided approaches provide end-to-end guarantees to individual flows with significantly less overhead than IntServ QoS like RSVP.

In terms of admission control, one can distinguish between parameter-based admission control (PBAC) and measurement-based admission control (MBAC) algorithms. Whereas the PBAC algorithms rely on a priori knowledge and accurate network traffic models to allot the resources, the MBAC algorithms rely on actual measurements and accurate estimation of QoS parameters. Brewer and Ayyagari [50] compare and analyse MPAC and PBAC in testbed evaluations. They conclude for bursty traffic patterns that the MBAC approach provides better network utilisation and a higher admission rate than the PBAC approach. Similar results from Mancuso and Neglia [51] prove true the superiority of MBAC algorithms about PBAC algorithms, in special for scenarios with bursty nature of self-similar flows. They discovered that MBAC algorithms make the system robust to statistical traffic properties. That is why we also decided to investigate more on MBAC algorithms and designed our approach according to this scheme. Moore [52] identified five characteristics for an appropriate MBAC algorithm. First a MBAC must provide a relationship between the traffic characteristic and the calibration control. Second, the estimator must incorporate the statistical nature of traffic. Third, the estimator and the MBAC must be matched to the task required. Fourth, the algorithms must be implementable with realistic resource requirements. Fifth, the policy performed by the MBAC influences the overall performance critically. Overall, he concludes that the correct maintenance of the current provision values is more important than the accuracy of short term traffic characterisation. Independently, we also designed our MBAC algorithm for QoSiLAN according to these principles and share the experiences. Jamin et al. [22] evaluated three MBAC and one PBAC algorithm in terms of performance for controlled load service. They configured the PBAC algorithm for capacity bounding. The three MBAC algorithms are based on equivalent bandwidth, acceptance region, and measured bandwidth. Although they do not aim on giving final conclusions on their simulation results, their evaluations reveal that a higher utilisation target than 80% causes packet loss in the network. In another survey Jamin and Shenker [53] observed that all known MBAC algorithms can be reduced to one formula, as shown in

$$\widehat{\nu} < f(\cdot)\,\mu - g(\cdot), \tag{5}$$

and be tuned with parameters to give the same result curves. In (5) $\widehat{\nu}$ is the measured load, μ is the link bandwidth, and $f(\cdot)$ and $g(\cdot)$ are functions of the source's reserved rate and the number of admitted sources. Therefore, they conclude and propose to focus future research on the tuning parameters, instead of the algorithms itself. Another observation is the structural limitations of MBAC algorithms. First, long lasting connections will statistically dominate the reservations over short lasting connections. Second, flows that traverse multihop paths have a higher risk of a rejected admission, if the switches perform the admission independently. For the QoSiLAN MBAC algorithm we considered the limitations and found solutions as described in the QoSiLAN policy

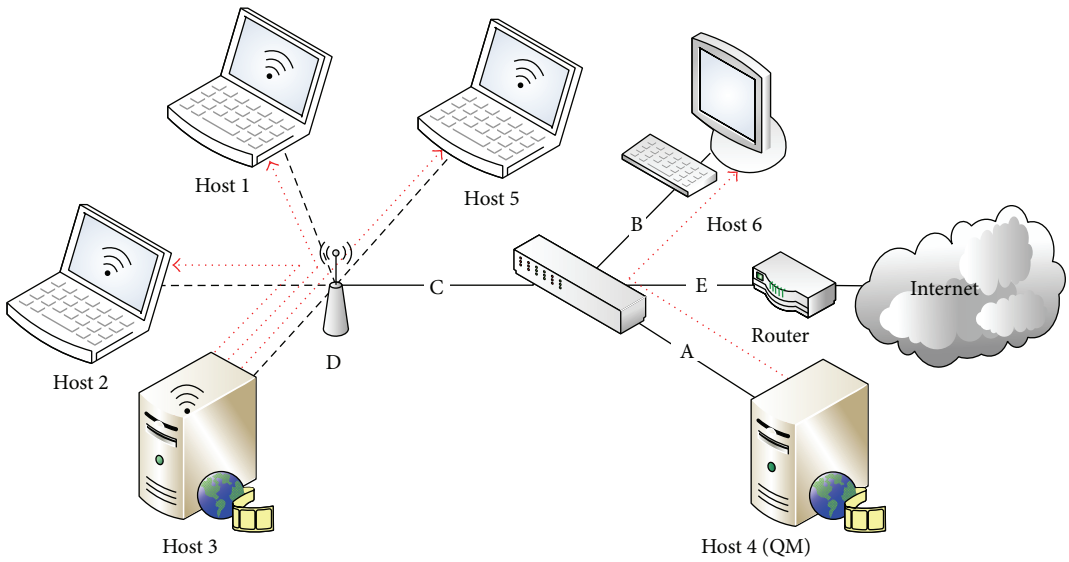


Figure 8: QoSiLAN evaluation scenario.

algorithm Section 2.5.2. The application of MBAC algorithms in the context of QoS for Quality of Experience (QoE) was shown by Latré and De Turck [54]. The authors propose a MBAC algorithm for provider based video rate controlling. They define policies of how providers can use MBAC algorithms and video rate control policing for the goal of revenue maximization or QoE. This is a passive approach to react on QoS degradation. Instead, the QoSiLAN framework aims on preventing congestion and interference traffic actively.

4. Materials and Methods

This section discusses the approaches, environments, and scientific methods used to achieve the presented results and evaluations as well as the research methodology to develop the proposed research solutions. Firstly, it presents the evaluation test-bed most evaluations were based on. Afterwards, for each proposed QoSiLAN framework key module, the special evaluation or simulation setup is described.

4.1. Home Scenario Evaluation Testbed. The evaluation test-bed was selected to represent a typical home environment with six active hosts, as presented in Figure 8. Three of the hosts are connected using fixed line 100Base-T Ethernet links and three hosts are connected using WiFi IEEE 802.11g links. In special, the simplex/duplex nature of the different link types and the hybrid QoS behaviour are represented by this setup. Each one wireless and one wired host are serving as media server and therefore as data source. The others are configured as media clients to consume media and demand for resources interactively. The Host 4 was configured as QoSiLAN Manager (QM), to manage the resources, since the intra-LAN traffic is in focus for this scenario. For setups, where the Internet to LAN traffic is in focus, the router is configured as QM. The hosts are Netbook devices with Windows 8 operating system. The router is a Linksys WRT-54GL device [55] running with the Linux based DD-WRT operating system. All devices, including the router, are equipped with the portable QoSiLAN framework stack. The test-bed network is isolated from the laboratory's traffic using the router's NAT and firewall functionality. The Internet link is routed through the laboratory LAN, sharing a 100 Mbit Internet link, provided by the facilities of the University of Applied Sciences Mittelhessen, Germany, which is connected to the German Scientific Network (DFN) backbone [56]. The DFN Internet backbone X-WIN is a science network, connecting more than 60 universities, science institutes, and science related companies within Germany, Europe, and abroad using one of the most powerful fiber-based communication networks in the world.

4.2. Link Layer Topology Discovery. The LLTD protocol was selected after intensive related work study [13, 14, 57] and comparison of existing topology discovery solutions [58–60]. It was found that the LLTD algorithm is the most appropriate state of the art approach of physical topology discovery in LANs. The protocol was implemented by Microsoft for their Windows operating system products since the release of Windows Vista, including a closed source LLTD Mapper service and LLTD Responder service for Microsoft Windows and an open source LLTD Responder for Linux based operating systems. The most important part, an API to the LLTD Mapper service or an open source implementation of it, is not available publicly. Although technical protocol descriptions exist and the algorithm was presented within a conference paper [57], a lot of implementation and algorithmic details of the LLTD mapping process are not published. So, to be able to use the technology within the QoSiLAN framework and for its evaluations the LLTD mapper had to be reverse engineered and reimplemented. This was a major task, since the mapping process is very complex and especially discovering deep segments and hosts is not trivial.

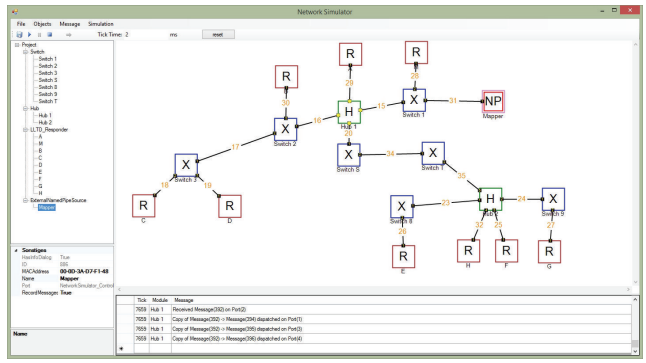


Figure 9: LLTD network simulator with the test topology loaded.

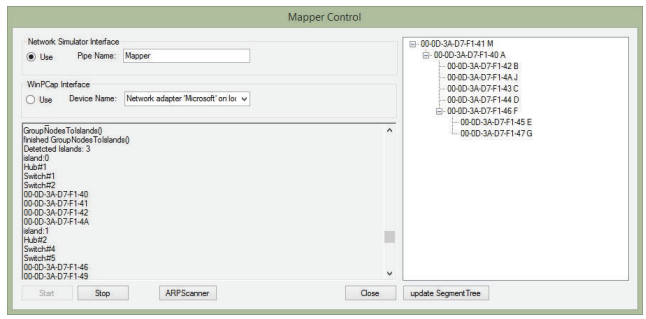


Figure 10: LLTD mapper application with the test topology's segment tree.

To reverse engineer the LLTD algorithm and to test the LLTD mapper during the implementation and reverse engineering phase, an Ethernet network simulator was developed, which simulates the behaviour of switches, hubs, and LLTD responder nodes. The network simulator, as shown in Figure 9 provides basic Ethernet functionality to emulate the Ethernet communication behaviour, addressing the switch's AIT building behaviour. In addition, the LLTD responder nodes also implement the LLTD responder behaviour for LLTD message sending and responding and the "sees" list. All addresses, tables, and lists are inspectable through the user interface. Additionally, the network simulator provides functionality to pause and continue the communication to provide rich debugging possibilities. It provides a live watch feature, to follow the Ethernet packet traversal through the LAN using animations. The LLTD mapper application, shown in Figure 10, is a separate component, which was developed to support both communication with the simulator through named pipes and portable Ethernet operation using the libpcap/winpcap API interfaces for Linux and Windows based systems [61, 62]. The LLTD mapper as well as the Ethernet Simulator was developed using the Microsoft .NET framework [63] and the Mono project framework [64] to support platform independence. For the other evaluation scenarios, which make use of the LLTD features and as final regression test, the LLTD mapper was tested and productively used within the evaluation test-bed, as presented in Figure 8. Further details on the work of the LLTD are not included in this paper, since we did not enhance this technology but only used it for the QoSiLAN framework and its test-beds.

4.3. Enhanced Statistical Protocol Identification. Preceded by intensive related work study, the SPID algorithm was identified as best fitting for the QoSiLAN framework since it fulfils the requirements of the targeted environment. It is light weight of high precision and enables protocol identification and application payload identification at the same time, even for encrypted or compressed traffic. After related work study and first tests the SPID algorithm was selected as appropriate base. The published implementation was evaluated to be not well implemented in terms of performance and memory usage and needed to be reimplemented. In addition, laboratory test showed that the SPID performance was not as good as expected. Therefore, all measures and parameters were evaluated and optimised. In addition, the measures were reconfigured and additional once developed and tested, to find a better measure configuration for the QoSiLAN framework evaluation test-bed. The completely reengineered algorithm was called eSPID. The eSPID implementation is written in C++ using the libpcap/winpcap API interfaces [61, 62].

The eSPID evaluations to find the best measure and algorithm configuration were carried out using a set of 3135 flows from 17 different protocols. The set of evaluation flows was recorded from real web-browsing and application usage under various usage scenarios, to cover different protocol

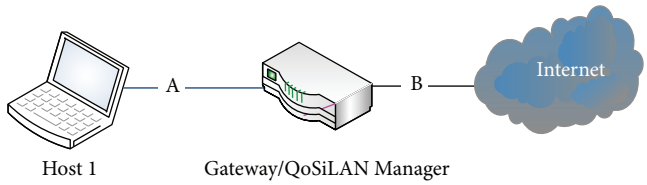


Figure 11: QoSiLAN evaluation scenario.

behaviours for the same protocol. During the development process and for the fine-tuning of the algorithm, the different measures were tested individually to verify their performance and usefulness for the whole set of measures.

4.4. Statistical Class-Based Bandwidth Prediction. The SCBP algorithm was developed from the motivation to have a simple and light-weight algorithm with low computing complexity. After literature research and investigation of real-world traffic from major video and audio streaming portals it was found that the state-of-the-art literature solutions do not handle the characteristics of nowadays Internet media streaming traffic explicitly. For that reason, the SCBP algorithm was designed from the practical observation that streams need a prior classification and case by case handling before predications should be applied. In particular, the first ten seconds of transmission were found as significant for the overall transfer behaviour. Dependent on the characteristic of the first ten second transfer behaviour, a systematic deviation from the expected results could be discovered. Therefore, intensive evaluations were carried out to optimise the classified results using individual correction factors for each class. In addition, different optimisation approaches were followed in parallel and compared to find the best optimisation set for the prediction results. All evaluations for the SCBP were performed with real Internet traffic from common WebTV, IPTV, Internet Radio, and on-demand platforms, located in Germany, United Kingdom, France, and the United States of America. The evaluation was carried out according to Figure 11 on Host 1. Host 1 is connected using 100BaseT Ethernet to an Internet gateway, which provides access to the University Of Applied Sciences Mittelhessen's (THM) Internet connection. The SCBP implementation is written in portable C++ using the libpcap/winpcap API interfaces [61, 62]. The streams were automatically identified using the eSPID algorithm [15] and classified according to the algorithm discussed in Section 2.2. Only streams with a minimum transfer time of 60 s were included in the evaluation, which resulted in a set sized of 463 samples. This allowed us to get the significant average bandwidth consumption value for the first 60 s ($\overline{B}_{60}$) for each flow. The ($\overline{B}_{60}$)-value served as reference and was used to validate the prediction accuracy after ten seconds (A_{10}).

4.5. QoS Signalling Layer Protocol for Local Area Networks. The QSLP-LAN was designed after intensive literature research and state-of-the-art Internet protocol specification investigations. The latest IETF recommendations were followed very strictly to specify the new protocol accordingly for the novel QoSiLAN's QoS model. The evaluations also include a critical quantitative analytical view on the IETF's NSIS specification and the overhead it causes in a distributed scenario. The NSIS-based QSLP-LAN implementation is written in C++ using the ACE ACE-Wrapper communication framework for a light-weight, portable multithreading and socket communication support.

The QSLP-LAN was evaluated within the test-bed presented in Figure 8. The communication was tested on the Windows based hosts, as well as on the DD-WRT enabled router. The Wireshark- [65] and the Tcpdump-tools [66] were used to inspect and analyse the signalling data. The QSLP-LAN was also assessed indirectly within the evaluations for the policing and admission control function, presented in Section 2.5.3. Additionally, the analytical evaluation was performed using mathematical calculations and projections assisted by Microsoft's Excel [67] application.

4.6. Policing and Admission Control. The policing and admission control algorithm was designed according to the best-practices found in the literature and in existing systems. The literature research helped to identify the crucial parts and to select the appropriate approach. The concrete implementation of the algorithm was designed by employing the features of the different QoSiLAN framework's modules and to use them efficiently. The policing and admission control module is written in C++ employing APIs to all other module and additionally accessing the Microsoft Windows Traffic Control API [68] and the Linux netfilter-tc [68] tools for active traffic management and control. The intermodule communication is realised using local socket communication.

The policing and admission control evaluations show the integration of the whole QoSiLAN framework, its behaviour, and the interaction of the individual key components. Therefore, the evaluation uses the complete scenario, as shown in Figure 8. A complete resource reservation procedure was selected, which drives the network in an overloading situation with QoS degradation of the application streams with the QoS demands. The evaluation setup was also designed to assess the MBAC and policing algorithm in an demonstrative way.

5. Results and Discussion

The evaluations were carried out in a real test-bed with Netbook hosts, connected using IEEE 802.11g and IEEE 802.3 100-BaseT Ethernet links, as shown in Figure 12. The presented scenario includes four wireless hosts, two fixed line hosts, an access point, a switch, and a router, which provides the Internet connection.

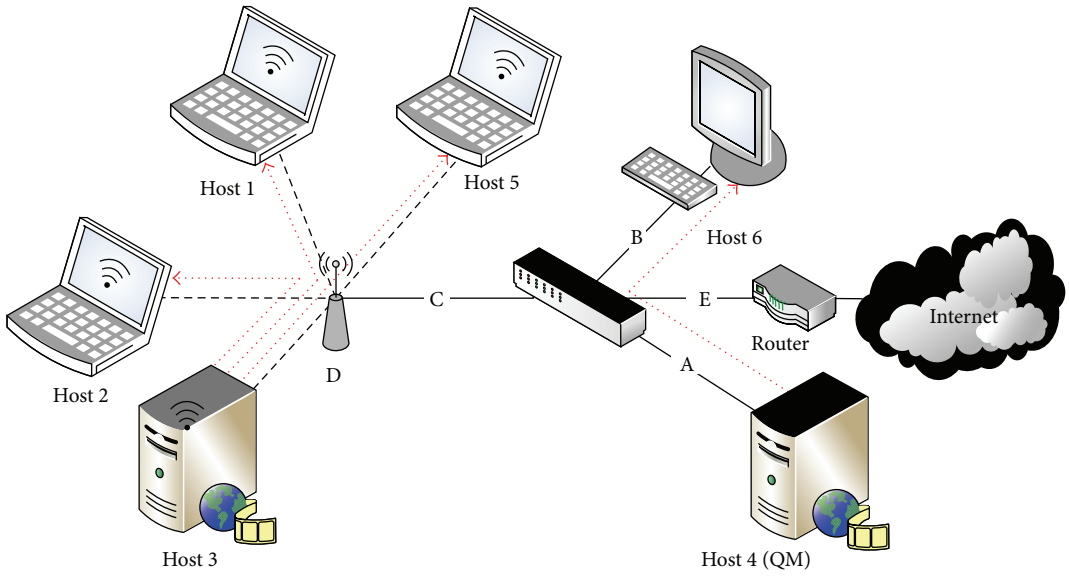


Figure 12: QoSiLAN evaluation scenario.

5.1. QoS Admission Control and Policing. To evaluate the concept of the QoSiLAN framework and the effectiveness of its admission and policing algorithms, the scenario shown in Figure 12 was configured to overload the network for proofing the system in a critical situation. In addition, the features of hybrid inter-access-medium QoS, per link bandwidth reservation, and simplex/duplex handling are shown within this example. To show the effectiveness of the QoS approach, Figure 13 depicts a direct comparison of a bandwidth conflict scenario with and without QoS enabled.

5.1.1. QoSiLAN Framework Evaluation. The effectiveness of the QoSiLAN's QoS approach of reserving bandwidth within the LAN by cooperative traffic shaping is illustrated in Figure 13.

This comparative evaluation was executed within the test-bed shown in Figure 12. Host 1, Host 2, and Host 5 request TCP streams with 8 Mbps from Host 3. In Figure 13(a) the results are presented, where there is no QoS configured, whereas in Figure 13(b) the results were generated with QoSiLAN's QoS functionality enabled and requested for all of the streams. All stream have a bandwidth demand of 3 Mbps. Since they all are sharing the same access medium, which was assessed with a capacity of 20.6 Mbps before a bandwidth conflict occurs when all streams are running at the same time. In Figure 13(b), the conflict is resolved by QoSiLAN, which detects the overprovisioning and hence does not admit the reservation request for Stream 3. Therefore, it limits Stream 3 to 3 Mbps. First, Host 1 requests Stream 1 and can start without disturbance. After 20 s Host 2 requests Stream 2. Since the access medium capacity is not exceeded in both scenarios, no problem occurs. The main difference until that point in time is the visible effect of the traffic shaper, which harmonises the variance of the bandwidth consumption. After 40 s Host 5 requests Stream 3. Figure 13(a) shows that this harms the transmission of Stream 2, which shares the residual bandwidth with Stream 3, now. Also Stream 1 is affected and disturbed in transmission. Figure 13(b) shows the QoSiLAN approach's effect, where the start of Stream 3 does not harm the transmission of the other streams. Only in the beginning of Stream 3 there is a short disturbance of Stream 2 until the traffic shaper in Host 5 is applied after 20 packets of flow recognition and identification.

The whole QoSiLAN framework was also evaluated using the test-bed in Figure 12. There, Host 4 was setup as QoSiLAN Manager. Table 2 shows a tabular view on the actions performed within the evaluation.

Before the start of the evaluation, Host 4 already performed the LLTD mapping process and assessed the wireless link TCP throughput from the wireless Host 3 to the other wireless Host 1 with $\mu_l = 10.6$ Mbps. This is a realistic throughput for IEEE 802.11g connections with both nodes connected to the same wireless access point, due to the simplex nature of the access medium. The tested throughput from the fixed Host 6 to the wireless Host 1 was measured to be 20.6 Mbps.

The maximum TCP throughput between the fixed Host 4 and Host 6 was tested with 87.7 Mbps. The tests were carried out under congestion free network conditions and a close physical link for the wireless hosts, using the iperf TCP and UDP bandwidth performance measurement tool [10]. In the beginning, the network is in idle state. For our evaluations, Host 1 starts requesting a video stream (Stream 1) from Host 4 with an average bandwidth of 8 Mbps, as shown in Figure 14. After 20 packets the eSPID module identified the stream type as video and after 10 s the average output bandwidth p_b^1 of Stream 1 was predicted with 8.0 Mbps. The admission control algorithm decided to admit the reservation, since it is in the rage of 80% link capacity 8.48 Mbps $= \mu_l \geq \mu_c = 8.0$ Mbps of the wireless link D. The QM initiated the QoS signalling and advised all wireless hosts to shape their outgoing traffic to other wireless hosts to $\mu_r = 2.12$ Mbps

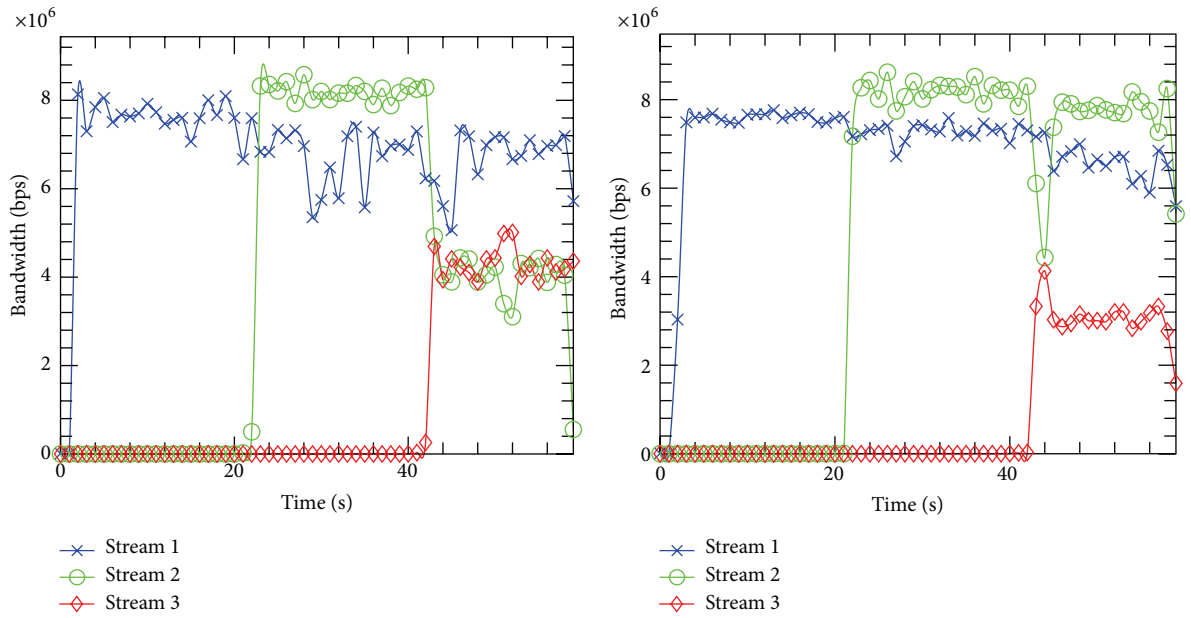


FIGURE 13: Conflict evaluation with and without QoSiLAN QoS enabled.

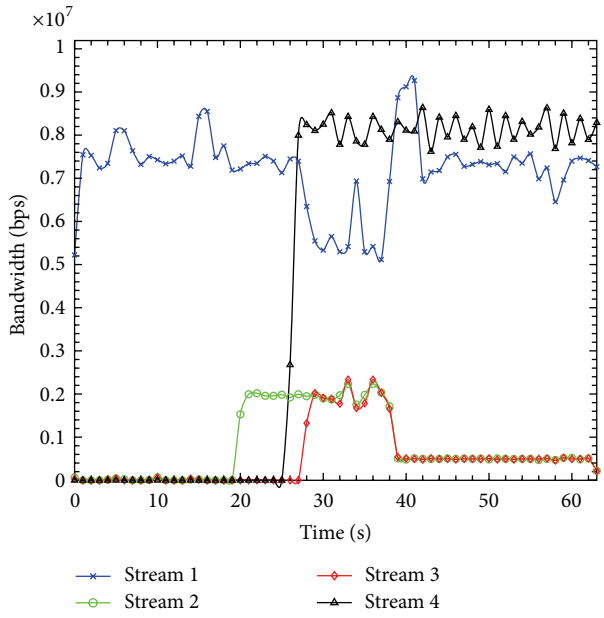


FIGURE 14: QoSiLAN evaluation.

residual bandwidth. Exactly 20 s after start, Host 2 also requests TCP data (Stream 2) from Host 3 with a data rate of 2 Mbps, which was predicted with $p_b^2 = 2.1$ Mbps. This stream was not identified as audio/video stream and therefore no reservation was initiated. The predicted bandwidth of Stream 2 (p_b^2) is smaller than the residual bandwidth μ_r and thus causing no QoS problems. Although for this flow no reservation is requested, the QM is informed about the current bandwidth occupation.

Now, both streams are running in parallel without any disturbance or interference. When Host 5 also requests 2.1 Mbps TCP traffic (Stream 3) 27 s after start, the wireless link capacity is exceeded and Stream 1 is disturbed significantly, as shown in Figure 14. This situation represents an unmanaged state, where no QoS is applied. This situation lasts for 6 s. As soon as Host 3 detects the significant outgoing traffic, it informs the QM about its amount. The QM detects the conflict and advises all wireless hosts to shape their outgoing traffic to other wireless targets to 500 Kbps. This happens in second 33, where the new policies are applied. Now, the QoS protection for Stream 1 is adapted again and Stream 1 returns to its desired throughput level. For targets

Table 2: Evaluation action schedule.

Time (sec)	Source host	Target host	Action
−10	4	4	LLTD mapping
−5	4	Any	Performance measurement
0	1	4	Request Stream 1
0	4	1	Start of Stream 1
2	4	Local	SPID identification [Stream 1: Video]
10	4	Local	SCBP prediction (Stream 1)
10	4	4	QoSiLAN_Req (Stream 1)
10	4	Any	QoSiLAN_Req (Stream 1)
19	2	3	Request Stream 2
19	3	2	Start of Stream 2
20	3	Local	SPID identification [Stream 2: None]
29	3	Local	SCBP prediction (Stream 2)
29	3	4	QoSiLAN_Status (Stream 2)
27	5	3	Request Stream 3
27	3	5	Start of Stream 3
28	3	Local	SPID identification [Stream 3: Video]
37	3	Local	SCBP prediction (Stream 3)
37	3	4	QoSiLAN_Req (Stream 3)
37	4	Any	QoSiLAN_Req (Stream 3)
37	any	Local	Apply new policy

outside the wireless link D a shaping limit of 1 Mbit was communicated. As soon as this new policy is applied by the hosts, the throughput of Stream 1 recovers to its desired state, as depicted in Figure 14. In parallel, a video stream from Host 4 to Host 6 (Stream 4) with $p_b^3 = 8.3$ Mbps is detected. Since no reservations are applied to the links A and B the reservation is admitted and does not affect the other reservations. This new reservation is also not communicated to the wireless hosts, since the bottleneck bandwidth for the path from the wireless hosts to Host 6 is lower than the residual bandwidth on link B. Host 6 has no limits to Host 4, due to the duplex nature of the Ethernet links and no reservations apply in that direction. In addition, there is no need to communicate the Ethernet reservation to the wireless nodes, since their outgoing traffic limit is lower than the one on the Ethernet links. Host 4 limits its outgoing traffic for other flows than Stream 4 to $p_b^3 - \mu_r^A = 8.3$ Mbps − 70.16 Mbps = 61.86 Mbps according the policing rules. Table 3 shows the shaping policies as applied at the end of the evaluation, when all reservation states are active. As one can see, all host receive individual policies, according to their location in the network. In that way perlink reservation states are enforced.

5.1.2. Conflict Management. Critical situations occur, if there are not enough resources available in the LAN. In this case the admission control function decides on three traffic priority levels:

(i) general Internet access (low priority);

(ii) AV-streaming/IPTV (higher priority);

(iii) VoIP/IP telephony (highest priority).

In the case of two conflicting reservation requests, the first come first served policy is applied. So, the last incoming conflicting request is declined, unless it belongs to a higher prioritised traffic class. If a higher prioritised request is recognised, the QH owning the lower one is informed with a tear down request for its reservation state, in forms of a QoSiLAN_RESERVE message with the tear flag set. Also all other QHs in the network are informed to delete the lower prioritised reservation state and to establish the higher one. This enables the QH to inform the user about the loss of QoS for the running service.

5.1.3. Limitations. Since the QoSiLAN approach relies on client support, most of the possible reasons limiting the framework's effectiveness are caused by lacking QoSiLAN support by hosts:

(i) *One host, with lacking QoSiLAN support,* can be managed by the corresponding host. In cases if the sender is not QoSiLAN aware, on behalf, the traffic-receiver may tell the QM about the needed resources. Both end-systems are not QoSiLAN aware, but the gateway discovered real-time traffic between them; the QM can initiate the reservation request on behalf of it, depending on matching policies. This is a probable scenario, if one node belongs to the LAN and is not QoSiLAN aware, and the correspondent host is located outside the LAN. In this case a QoSiLAN aware gateway may detect the need of resource reservation, since it is part of the path between the hosts.

(ii) *Two hosts, with lacking QoSiLAN support,* inside the LAN communicating with each other would cause uncontrolled and unknown traffic to the LAN. This may harm the communication of other hosts, since there is a way neither to prohibit the traffic nor to even get knowledge of it in a switched network. So, LAN internal communication between two hosts not supporting QoSiLAN has to be avoided by the administrators.

(iii) *Data traffic, from or to hosts, which are not locatable by the LLTD protocol* would cause QoSiLAN to limit the bandwidth, in case of reservations, network wide and not per link. Since the data path in the LAN's topology is not traceable by the QM, the reservation must be operative to all physical links in the whole LAN. This may reduce the overall performance of the network. In addition, the QM has to calculate the reserved capacities in the network and to take care that it reserves not more resources than available. It has to mind a minimal bandwidth capacity for all hosts in the network to enable best effort for at least signalling applications.

Table 3: Traffic shaping policies.

Source	Host 1	Host 2	Host 3	Host 4	Host 5	Host 6
Target				Limit		
Host 1	—	500 Kbps	500 Kbps	1 Mbps	500 Kbps	500 Kbps
Host 2	500 Kbps	—	500 Kbps	1 Mbps	500 Kbps	1 Mbps
Host 3	500 Kbps	500 Kbps	—	1 Mbps	500 Kbps	1 Mbps
Host 4	1 Mbps	1 Mbps	1 Mbps	—	1 Mbps	no limit
Host 5	500 Kbps	500 Kbps	500 Kbps	1 Mbps	—	500 Kbps
Host 6	1 Mbps	1 Mbps	1 Mbps	61.86 Mbps	1 Mbps	—

Table 4: QoSiLAN request header sizes.

Header name	Size [bytes]
GIST	12
GIST NSLP DATA	4
QoS NSLP	4
QoS RSN	4
QoS EpochID	8
QoS Refresh Period	4
QoS Bound Session ID	36
QSpec Common Object	4
QSpec Common QSpec	4
QSpec Object	4
QSpec TMOD-1	24
QSpec Excess Treatment	8
QSpec RPP-IPv4	20
Sum	136

Table 5: QoSiLAN response header sizes.

Header name	Size [bytes]
GIST	12
GIST NSLP DATA	4
QoS NSLP	4
QoS RSN	4
QoS INFO SPEC	8
Sum	32

5.2. QoS Signalling Protocol. This section evaluates the quantity, scalability and performance of the protocol. A typical QoSiLAN_RESERVE message has a length of 124 bytes, including the UDP, IPv4, and GIST header. The GIST message header has a size of 12 bytes. The GIST NSLP data header has a length of 4 bytes. The GIST payload consists of a QoS message, which starts with a QoS NSLP header with a length of 4 bytes, followed by the QoS objects. All QoS objects start with a 2 byte common header. The minimum size is 4 bytes, additionally 32 bit fields are indicated by the length parameter at byte 2 of the QoS common object header. These are the RSN, EpochID, Refresh Period, Bound Session ID, and QSpec. The sizes of all headers are listed in Table 4. The QSpec contains QSpec objects with a Common QSpec header at the start. The QSpec objects start with a 4 byte parameter header, followed by the object's payload header. These are the TMOD-1, Excess Treatment, and the RPP-IPv4. The sum of headers gives a total message length of 116 bytes, as listed in Table 4.

The QoSiLAN_Response message has a size of 32 bytes, as listed in Table 5. The INFO_SPEC object consists of a 4 byte common header and the 4 byte ESI, which carries the IPv4 address of the message source. For each host in the network, the QM has to generate one message for each QH in the LAN, which leads to the network request message load (B_r), generated by the QM, as defined in

$$f(B_r) = (N_H - 1)^2 * s_{\text{req}} + (N_H - 1)^2 * s_{\text{rep}}. \quad (6)$$

There, N_H describes the number of host and s_{req} is the size of the request and s_{rep} the size of response messages, respectively. The messages from and to the QM itself are not routed to the network and therefore not relevant to the network load. Figure 15 shows the scalability problem of the proposed protocol. It shows the graphs of number of messages and the generated amount of data in one graph, scaled by the number of hosts in the network. As one can see, already 25 hosts in the network cause over 129,024 bytes of traffic occupied by 1,152 messages sent in the network for a single QoSiLAN reservation session establishment. The 1 MB mark is hit with 68 hosts causing 8,978 messages. The number of 213 hosts generates approximately 10 MB of traffic using 89,888 messages. Over 1 billion messages are needed for a LAN with 709 hosts, generating 112,283,136 bytes of data.

The QSLP-LAN was designed following the latest IETF recommendations for new signalling protocols. Since the QoSiLAN approach is designed only for small networks with not more than 25 hosts, the number of messages and data used for signalling is not regarded as relevant for the system. As expected, the signalling effort for a cooperative approach is very high. Anyway, the evaluation prooves that a redesign of the signalling procedure would be needed in case of higher scalability demands. An approach to reduce the number of message and the amount of data needed would be to pack the individual policy information affecting one host into one single message, using the maximum path MTU for the transfer. Also, a more minimalistic, proprietary message design with compression enabled would significantly save signalling resources. For example, a QoSiLAN_Request message could be stripped to 83 bytes instead of 136 bytes, when removing all structural NSIS information and keeping only the essential data, without data compression. This is reduction of approx. 39%. When using a MTU size of 1400 bytes, 16 messages would fit into one packet. As depicted in Figure 15, the

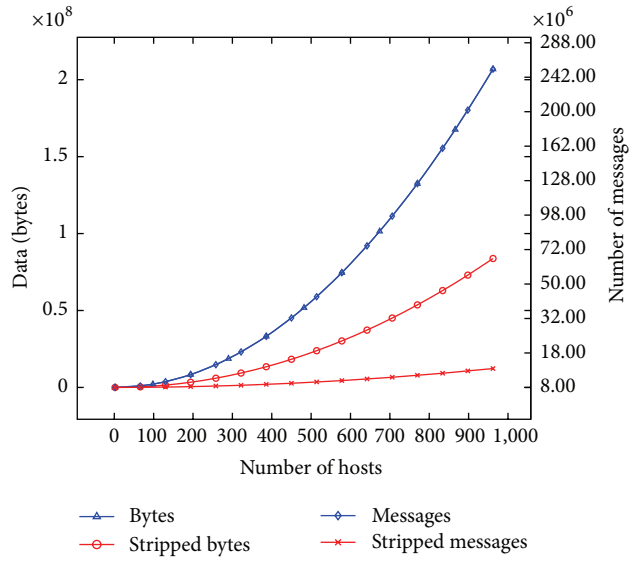


FIGURE 15: QoSiLAN signalling effort.

amount of data needed to to establish a QoS state in a network with 709 nodes would be reduced from 112,283,136 bytes to 45,493,248 bytes, which is a reduction of 59%. The number of message would be reduced by 94% from 1,002,528 to 59,472. In a 25-node network the number of messages would reduce from 1152 to 39 and the number of amount of data needed would scale down from 105,984 bytes to 41,864 bytes.

5.3. Statistical Class-Based Bandwidth Prediction. The evaluation helped to calibrate the algorithm and to validate its accuracy. To improve the readability of this section, Abbreviations provides a compact overview about the most important abbreviations. All evaluations were performed with real Internet traffic from common WebTV, IPTV, Internet Radio, and on-demand platforms, located in Germany, UK, France, and the USA. The evaluation was carried out according to Figure 11 on Host 6. The streams were automatically identified using the eSPID algorithm [15] and classified according to the algorithm discussed in Section 2.2. All streams had a minimum transfer time of 60 s. This allowed us to get the significant average bandwidth consumption value for the first 60 s ($\overline{B}_{60}$). This value served as reference and was used to validate the prediction accuracy after ten seconds (A_{10}):

$$A_{10} = \frac{P_{10}}{\overline{B}_{60}}. \tag{7}$$

Other important measures are the Prediction Hit Rate (PHR), the Mean Prediction Accuracy Ratio (MPAR), and the Over-Estimation Rate (OER):

(i) The Prediction Hit Rate specifies the number of predictions, which are in the range of $0.8 \preceq A_{10} \preceq 2$, as percentage. This range was specified to filter unacceptable outliers for consumer media traffic prediction.

(ii) The Mean Prediction Accuracy Ratio specifies the mean prediction accuracy $\overline{A}_{10}$ over all samples. Since the prediction causes network blocking through QoSiLAN reservations, the Mean Prediction Accuracy Ratio reveals the average network blocking rate.

(iii) The Over-Estimation Rate specifies the number of prediction accuracy ratios with $A_{10} > 1$ in percentage. This is an important measure for QoS critical applications, since it reflects the QoS assurance probability a prediction algorithm configuration can achieve.

For the evaluations a set of 1442 monitored media streams was used. All these parameters were evaluated for the whole stream set Class Less (CL) and per class individually Class-Based (CB). The k-value in (3) was optimised for each of these three measures individually to find the best method for optimisation as defined in the following:

(i) The k-value for Prediction Hit Rate was evaluated for the maximum Prediction Hit Rate in (3), to receive the maximum number of results in the acceptable range.

(ii) For the MPHR, the k-value was evaluated to give in (3) a result in average of 1, which reflects the best mean network blocking/utilisation rate.

(iii) The QoS optimization's k-value was evaluated to result in (3) an average of 90% Over-Estimation Rate to ensure a minimum of 90% QoS assurance.

Finding the best optimisation configuration is challenging. First, the Mean Prediction Accuracy Ratio should be near to 1, to ensure the best network utilisation. This would be reflected by traditional error calculation like the Root Mean Square Error (RMSE). Second, at the same time as much A_{10} results should match the acceptable range. Third, since the QoS is important, as much as possible A_{10} results should be greater than 1. To find the best optimisation aspects, which meet all

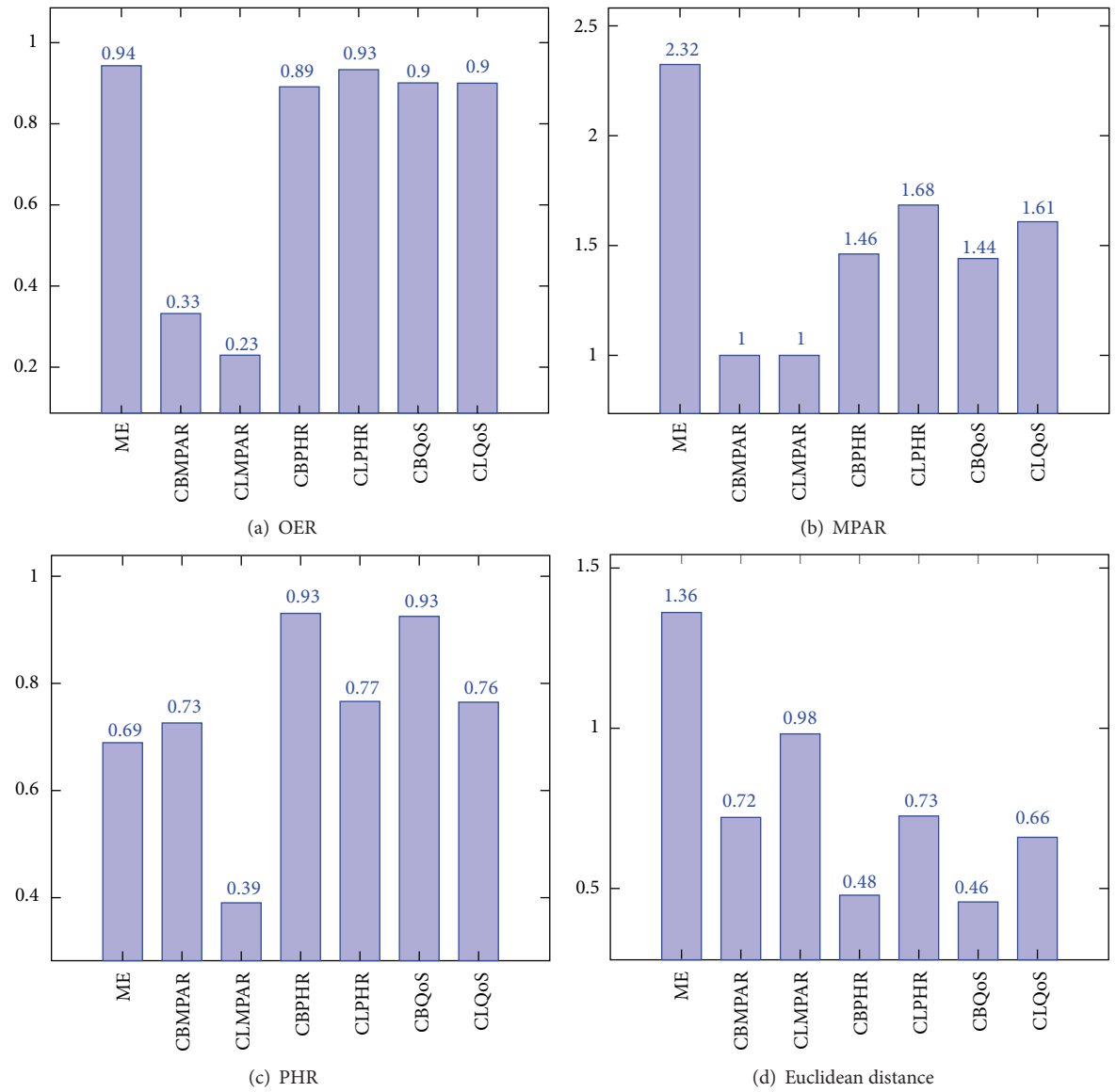


Figure 16: Performance assessment of optimisation results.

these requirements at the same time, the Euclidean distance (**d**) was employed, as shown in

$$\mathbf{d} = \sqrt{(1 - \mathrm{MPAR})^2 + (1 - \mathrm{PHR})^2 + (1 - \mathrm{OER})^2}. \quad (8)$$

It takes all three parameters into account and provides the minimum **d**-value, which reflects the best fitting algorithm configuration. In Figure 16(d) a smaller value of **d** indicates a better performance, taking into account the three measures MPAR, Prediction Hit Rate, and OER. Table 6 gives an overview about the results achieved for the different configurations. There, also the median ($\widetilde{\mathrm{MPAR}}$) and the mean accuracy ($\overline{\mathrm{MPAR}}$) values are listed, which enable a first assessment of the results. While the mean accuracy value emphasises the average network utilisation, the median accuracy value gives an indication about the distribution of values. Figures 16(a)–16(c) give an overview about the three different optimisation aspects, whereas Figure 16(d) shows the results after the Euclidean distance calculation, which reflects a combination of the previous ones. In Figure 16(a) the Over-Estimation Rate optimisation, with a target value of 1, indicates that the Mean Estimation (ME) and CLPHR aspects perform best. The CBQoS performs only on the third place together with CLQoS aspect. In Figure 16(b) the Mean Prediction Accuracy Ratio results with a target value of 1 that indicates the optimum for the CLMPAR and CBMPAR aspects, as expected, but the CBQoS aspect performs as next best. The Prediction Hit Rate aspect results in Figure 16(c) show the best performance for the CBPHR and the CBQoS results, which should be near to 1 to be the best. This is interesting and shows that the optimisation for Prediction Hit Rate without classes cannot provide an average ratio better than 0.77. The class-based approach gives a better

Table 6: Overview of results.

Optimization	k	$\widetilde{\text{MPAR}}$	**d**	$\overline{\text{MPAR}}$	PHR	PAR > 1	PAR < 1
CLME	1.0000	1.1561	1.3613	2.3241	68.93%	94.31%	5.69%
CBMPAR	Class based	0.8676	0.7218	1.0000	72.61%	33.22%	66.78%
CLMPAR	0.6335	1.2023	0.9824	1.0000	39.04%	22.95%	77.05%
CBPHR	Class based	1.1949	0.4790	1.4613	93.07%	89.11%	10.89%
CLPHR	1.0670	0.7324	0.7261	1.6842	76.63%	93.34%	6.66%
CBQoS	Class based	1.2335	0.4580	1.4408	92.51%	90.08%	9.92%
CLQoS	1.0189	1.1779	0.6597	1.6083	76.49%	90.01%	9.99%

density of results in the acceptable range. The Euclidean distance assessment in Figure 16(d) illustrates the advantage of the CBQoS approach, which does not perform best in all categories, but collectively over all optimisation aspects it shows the best performance.

The statistical class-based bandwidth prediction algorithm using the CBQoS optimisation configuration was proved to perform best for predicting bandwidth requirements for a relative long period of 60 s. On the first view the classless results look also very good and perform in some configurations similar to the class-based approach or even better. Finally the Euclidean distance comparison in Figure 16(d) reveals the advantage of the CBQoS optimisation approach, since it brings the deviation of results into account, combining the different measures. Also, we showed, as presented in Figure 16, that a 100% prediction accuracy with a minimum error is not desirable, as it causes a higher underestimation probability than, for example, an QoS optimisation case. Generally, it is more critical for the media streams, if their resources are underprovisioned, since they are not properly protected from congestion then. In contrast, blocking of more resources than needed is critical for the overall network performance and utilisation. In the application of QoS, a little overestimation of resources is also to be regarded as positive, since this allows the streams to prebuffer faster at the receiver, which results in more robustness against the variability of network performance and provides a better stream isolation against disturbing traffic and congestion.

6. Conclusion

In this paper we presented and evaluated the QoSiLAN framework in its whole for the first time. The interaction of the different key technology, which enable link based resource reservation by autonomous policing and admission control for QoS in LANs, was shown. The QoSiLAN admission control and policing algorithm was designed to take care not to block more resources in the network than needed, by allowing to react dynamically on reservation violations and network congestion in an appropriate manner autonomously. The QoSiLAN framework was presented to be an effective method to reserve bandwidth on individual links in LANs without the network's QoS assistance and therefore to provide self-organised QoS for unmanaged networks. The admission control and policing function in the QM utilizes the QH information from the topology mapping and the traffic-analysis and -reporting to take informed decisions for managing the LAN's resources efficiently. The new SCBP algorithm was presented along with detailed evaluations to optimise it for the best resource predication performance within the QoSiLAN framework. The evaluations show that QoS conflicts are detected reliably and a solution is enforced in an autonomous way. The novel QoSiLAN QoS model, designed to enforce cross-access-technology link-based bandwidth reservation by collaborative traffic shaping without network assistance using the dedicated QSLP-LAN protocol, was proved within the evaluations. The QoSiLAN framework is implementable in a lightweight manner. It does not rely on network support, but on host collaboration. In addition, not all hosts in the network need to support the framework essentially. For a proper operation, the support by at least one of the communication parties within the LAN is required, if the router is QoSiLAN aware. Traffic sources like Media-NAS devices and the Internet router may implement the framework preferably to allow for an optimised operation. This makes the QoSiLAN framework easy to implement and to be deployable in realistic scenarios. The autonomous configuration and operation features qualify the framework for nonexpert deployment and application.

Abbreviations

OER: Over-Estimation Rate
ME: Mean Estimation
PHR: Prediction Hit Rate
MPAR: Mean Prediction Accuracy Ratio
CBQoS: Class-Based Quality of Service
CBPHR: Class-Based Prediction Hit Rate
CBMPAR: Class-Based Mean Prediction Accuracy Ratio
CLQoS: Class Less Quality of Service
CLPHR: Class Less Prediction Hit Rate
CLMPAR: Class Less Mean Prediction Accuracy Ratio.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] Cisco. Cisco data meter, 2013, http://www.ciscovni.com/data-meter/index.html.

[2] Cisco, "Cisco Visual Networking Index: Forecast and Methodology, 2012–2017," white paper, 2013.

[3] Qualcomm, "Qualcomm Introduces StreamBoost Technology to Optimize Performance and Capacity of Home Networks," 2013, https://www.qualcomm.com/news/releases/2013/01/04/qualcomm-introduces-streamboost-technology-optimize-performance-and.

[4] R. Braden, L. Zhang, S. Berson, S. Herzog, and S. Jamin, "Resource ReSerVation Protocol (RSVP)—version 1 functional specification," Proposed Standard RFCs 2205, 2750, 3936, 4495, 5946, 6437, 6780, 1997, http://www.ietf.org/rfc/rfc2205.txt.

[5] M. Cooperation, "Rsvp service," 2013, http://msdn.microsoft.com/en-us/library/windows/desktop/aa374144(v=vs.85).aspx.

[6] ITU, *A Generic Home Network Architecture with Support for Multimedia Services (Recommendation ITU-T H.622)*, International Telecommunications Union, 2008.

[7] U. Forum, "Upnp-qos architecture v3, for upnp version 1.0," 2014, http://upnp.org/specs/qos/UPnP-qos-Architecture-v3.pdf.

[8] A. M. J. Manner and G. Karagiannis, Nslp for quality-ofservice signaling (qos nslp), internet draft (draft-ietf-nsisqos-nslp-16), work in progress, 2009, http://tools.ietf.org/html/draft-ietf-nsis-qos-nslp-16.

[9] S. Akhshabi, A. C. Begen, and C. Dovrolis, "An experimental evaluation of rate-adaptation algorithms in adaptive streaming over HTTP," in *Proceedings of the 2nd Annual ACM Multimedia Systems Conference (MMSys '11)*, pp. 157–168, February 2011.

[10] iperf, "Tcp and udp bandwidth performance measurement tool," 2013, https://github.com/esnet/iperf.

[11] R. Black, A. Donnelly, and C. Fournet, "Ethernet topology discovery without network assistance," in *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP '04)*, pp. 328–339, IEEE, Berlin, Germany, October 2004.

[12] M. Corporation, "Lltd and qos for media experiences," 2013, http://msdn.microsoft.com/library/windows/hardware/gg463099/.

[13] M. Corporation, "[MS-LLTD]: link layer topology discovery (LLTD) protocol specification," Open Specifications Documentation, 2009, http://msdn.microsoft.com/en-us/library/cc233983%28PROT.10%29.aspx.

[14] Y. Sun, Z. Wu, and Z. Shi, "The physical topology discovery for switched ethernet based on connections reasoning technique," in *Proceedings of the International Symposium on Communications and Information Technologies (ISCIT '05)*, pp. 44–47, IEEE, October 2005.

[15] C. Köhnen, C. Überall, F. Adamsky, V. Rakočević, M. Rajarajan, and R. Jäger, "Enhancements to statistical protocol identification (SPID) for self-organised QoS in LANs," in *Proceedings of the 19th International Conference on Computer Communications and Networks (ICCCN '10)*, pp. 1–6, IEEE, Zürich, Switzerland, August 2010.

[16] E. Hjelmvik and W. John, "Statistical protocol identification with SPID: preliminary results," in *Proceedings of the Swedish National Computer Networking Workshop*, 2009, http://spid.sourceforge.net/sncnw09-hjelmvik_john-CR.pdf.

[17] A. S. Incorporated, "Http dynamic streaming (HDS)," 2013, http://www.adobe.com/de/products/hds-dynamic-streaming.html.

[18] J. Manner, R. Bless, J. Loughney, and E. Davies, "Using and Extending the NSIS Protocol Family," RFC 5974 (Informational), 2010, http://www.ietf.org/rfc/rfc5978.txt.

[19] J. Manner, G. Karagiannis, and A. McDonald, "NSIS Signaling Layer Protocol (NSLP) for quality-of-service signaling. (Experimental)," RFC 5974, 2010, http://www.ietf.org/rfc/rfc5974.txt.

[20] H. Schulzrinne and R. Hancock, "GIST: general internet signalling transport," RFC 5971, 2010, http://www.ietf.org/rfc/rfc5971.txt.

[21] G. Ash, A. Bader, C. Kappler, and D. Oran, "QSPEC Template for the Quality-of-Service NSIS Signaling Layer Protocol (NSLP)," RFC 5975 (Experimental), 2010, http://www.ietf.org/rfc/rfc5975.txt.

[22] S. Jamin, S. J. Shenker, and P. B. Danzig, "Comparison of measurement-based admission control algorithms for controlled-load service," in *Proceedings of the 16th Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution (INFOCOM '97)*, vol. 3, pp. 973–980, IEEE, Kobe, Japan, April 1997.

[23] D. Hock, N. Bayer, R. Pries et al., "QoS provisioning in WLAN mesh networks using dynamic bandwidth control," in *Proceedings of the 14th European Wireless Conference (EW '08)*, pp. 1–7, IEEE, Prague, Czech Republic, June 2008.

[24] M. Louvel, A. Plantec, and J.-P. Babau, "Resource management for multimedia applications, distributed in open and heterogeneous home networks," *Journal of Systems Architecture—Embedded Systems Design*, vol. 59, no. 3, pp. 121–134, 2013.

[25] M. Louvel, P. Bonhomme, J.-P. Babau, and A. Plantec, "A network resource management framework for multimedia applications distributed in heterogeneous home networks," in *Proceedings of the 25th IEEE International Conference on Advanced Information Networking and Applications (AINA '11)*, pp. 724–731, Singapore, March 2011.

[26] M. Louvel, J. Pulou, A. Plantec, and J.-P. Babau, "Quantity of resource aggregation for heterogeneous resource reservation for multimedia applications," in *Proceedings of the 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA '10)*, pp. 1–4, IEEE, Bilbao, Spain, September 2010.

[27] S. Ghahramani, *Fundamentals of Probability: With Stochastic Processes*, Pearson/Prentice Hall, 2005, http://books.google.ca/books?id=M1GUQgAACAAJ.

[28] I. R. Tools, "tc—show/manipulate traffic control settings," 2014, http://www.linuxmanpages.com/man8/tc.8.php.

[29] J.-P. Laulajainen and M. Hirvonen, "Automatic QoS control in UPnP home networks," in *Proceedings of the IEEE Symposium on Computers and Communications (ISCC '09)*, pp. 455–460, IEEE, Sousse, Tunisia, July 2009.

[30] V. Suraci, G. Oddi, N. Mattiacci, and A. Angelucci, "Admission control and drop strategies in a UPnP-QoS controlled home network," in *Proceedings of the IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC '10)*, pp. 2811–2816, IEEE, Instanbul, Turkey, September 2010.

[31] M. Castrucci, G. Oddi, G. Tamea, and V. Suraci, "Application QoS management and session control in a heterogeneous home network using Inter-MAC layer support," in *Proceedings of the Future Network and Mobile Summit*, pp. 1–9, Florence, Italy, June 2010.

[32] J.-L. Chen, M.-C. Chen, and Y.-R. Chian, "QoS management in heterogeneous home networks," *Computer Networks*, vol. 51, no. 12, pp. 3368–3379, 2007.

[33] L. Westberg, M. Jacobsson, M. de Kogel et al., "Resource management in diffserv on demand (roda) phr, internet

draft (draft-westberg-rmd-od-phr-04), work in progress," 2003, http://tools.ietf.org/html/draft-westberg-rmd-od-phr-04.

[34] H. Y. Lee, S. T. Moon, and J. W. Kim, "Enhanced UPnP QoS architecture for network-adaptive streaming service in home networks," *IEEE Transactions on Consumer Electronics*, vol. 53, no. 3, pp. 898–904, 2007.

[35] L. Brewka, P. Sköldström, J. Nelis, H. Wessing, and C. Develder, "Automatic provisioning of end-to-end QoS into the home," *IEEE Transactions on Consumer Electronics*, vol. 57, no. 4, pp. 1670–1678, 2011.

[36] ITU, *Architecture of MediaHomeNet (Recommendation ITU-T J.190)*, International Telecommunications Union, Geneva, Switzerland, 2nd edition, 2007.

[37] Home Gateway Initiative, "Home gateway technical requirements, release 1," Tech. Rep., Home Gateway Initiative, 2006.

[38] The Architecture and Transport Working Group and DSL Home Technical Working Group, "Multiservice delivery framework for home networks," DSL Forum, Technical Report, 2004.

[39] A. Y. Haikal, M. Badawy, and H. A. Ali, "Towards internet QoS provisioning based on generic distributed QoS adaptive routing engine," *The Scientific World Journal*, vol. 2014, Article ID 694847, 29 pages, 2014.

[40] G. Apostolopoulos, S. Kama, D. Williams, R. Guerin, A. Orda, and T. Przygienda, "QoS Routing Mechanisms and OSPF Extensions," RFC 2676 (Experimental), 1999, http://www.ietf.org/rfc/rfc2676.txt.

[41] Object Management Group, "Data distribution service for real-time systems," Object Management Group, 2007, http://www.omg.org/.

[42] A. Al-Roubaiey and M. A. R. Alkhiaty, "Qos-aware middleware for ubiquitous environment: a review and proposed solution," *Journal of Computational Engineering*, vol. 2014, Article ID 725960, 7 pages, 2014.

[43] G. Sivaradje and P. Dananjayan, "Dynamic resource allocation algorithm for next generation wireless multimedia. Services," in *Proceedings of the 8th IEEE International Conference on Communication Systems (ICCS '02)*, vol. 2, pp. 752–754, IEEE, Singapore, November 2002.

[44] K. Papagiannaki, N. Taft, Z.-L. Zhang, and C. Diot, "Long-term forecasting of Internet backbone traffic," *IEEE Transactions on Neural Networks*, vol. 16, no. 5, pp. 1110–1124, 2005.

[45] R. H. Filho and J. E. B. Maia, "Network traffic prediction using PCA and K-means," in *Proceedings of the 12th IEEE/IFIP Network Operations and Management Symposium (NOMS '10)*, pp. 938–941, IEEE, Osaka, Japan, April 2010.

[46] S. Chong, S.-Q. Li, and J. Ghosh, "Predictive dynamic bandwidth allocation for efficient transport of real-time VBR video over ATM," *IEEE Journal on Selected Areas in Communications*, vol. 13, no. 1, pp. 12–23, 1995.

[47] H. Feng, Y. Shu, S. Wang, and M. Ma, "SVM-based models for predicting WLAN traffic," in *Proceedings of the IEEE International Conference on Communications (ICC '06)*, pp. 597–602, Istanbul, Turkey, July 2006.

[48] Q. He, C. Dovrolis, and M. Ammar, "On the predictability of large transfer TCP throughput," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM '05)*, pp. 145–156, ACM, Philadelphia, Pa, USA, August 2005.

[49] M. Yang, Y. Huang, J. Kim, M. Lee, T. Suda, and M. Daisuke, "An end-to-end QoS framework with on-demand bandwidth reconfiguration," in *Proceedings of the 18th IEEE Annual Workshop on Computer Communications (CCW '03)*, pp. 66–74, IEEE, Dana Point, Calif, USA, October 2003.

[50] O. T. Brewer and A. Ayyagari, "Comparison and analysis of measurement and parameter based admission control methods for Quality of Service (QoS) provisioning," in *Proceedings of the IEEE Military Communications Conference (MILCOM '10)*, pp. 184–188, San Jose, Calif, USA, November 2010.

[51] V. Mancuso and G. Neglia, "Performance improvements on self-similar traffic using measurement-based admission control," 2002, http://www-sop.inria.fr/members/Vincenzo.Mancuso/MNB02.pdf.

[52] A. W. Moore, *Measurement-based management of network resources [Ph.D. thesis]*, Corpus Christi College, University of Cambridge, Cambridge, UK, 2002, http://www.cl.cam.ac.uk/~awm22/publications/moore2002phd.pdf.

[53] S. Jamin and S. Shenker, "Measurement-based admission control algorithms for controlled-load service: a structural examination," Tech. Rep. CSE-TR-333-97, University of Michigan, Ann Arbor, Mich, USA, 1997.

[54] S. Latré and F. De Turck, "Joint in-network video rate adaptation and measurement-based admission control: algorithm design and evaluation," *Journal of Network and Systems Management*, vol. 21, no. 4, pp. 588–622, 2013.

[55] T.F.E. Wikipedia, "Linksys wrt54g series," 2015, http://en.wikipedia.org/wiki/Linksys_WRT54G_series.

[56] D.F.N.G.S. Network, "Das wissenschafftsnetz xwin," 2015, http://www.dfn.de/xwin.

[57] R. Black, A. Donnelly, and C. Fournet, "Ethernet topology discovery without network assistance," in *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP '04)*, pp. 328–339, IEEE, Berlin, Germany, October 2004.

[58] Y. Bejerano, Y. Breitbart, M. N. Garofalakis, and R. Rastogi, "Physical topology discovery for large multisubnet networks," in *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM '03)*, vol. 1, pp. 342–352, IEEE, March-April 2003.

[59] S. S. Ands, "Discovering internet topology," 1998, http://www.cs.cornell.edu/skeshav/papers/discovery.pdf.

[60] M. Rose, "Management Information Base for network management of TCP/IP-based internets: MIB-II. RFC 1158 (Proposed Standard)," Obsoleted by RFC 1213, 1990, http://www.ietf.org/rfc/rfc1158.txt.

[61] NRG, "Libpcap—the packet capture library," http://sourceforge.net/projects/libpcap/.

[62] D.C. CACE Technologies, "Winpcap: The windows packet capture library," http://www.winpcap.org/.

[63] Microsoft Corporation, "The microsoft net framework," 2002, http://www.microsoft.com/net.

[64] Mono Project, "Cross platform, open source .net framework," 2004, http://www.mono-project.com/.

[65] Wireshark Foundation, "Wireshark," 2015, https://www.wireshark.org/.

[66] Tcpdump, "Tcpdump, a powerful command-line packet analyzer," 2015, http://www.tcpdump.org/.

[67] Microsoft Corporation, "Excel application," 2015, https://products.office.com/en-us/Excel.

[68] Microsoft Corporation, "Traffic control application programming interface (TC API)," 2015, https://msdn.microsoft.com/en-us/library/windows/desktop/aa374468%28v=vs.85%29.aspx.